'KHASI BIBLE KONKORDANT

A project report submitted in partial fulfillment of the requirements for the degree
Of

Bachelor of Computer Applications (BCA)
OF
North Eastern Hill University (NEHU)
2017

Submitted by

Name	:	**MACDONALD KHARSAHNOH**
Roll No.	:	**P1500070**
Reg. No.	:	**14534 of 2013-14**

DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS
SHILLONG COLLEGE
SHILLONG 793003
MEGHALAYA.

CERTIFICATE

This is to certify that the Project work titled ***'KHASI BIBLE KONKORDANT'*** is a bonafide work done by ***MACDONALD KHARSAHNOH,*** University Roll No.: ***P1500070,*** Registration No.: ***14534 of 2013-14,*** under my guidance during the final year of the course.

Sir. Ransly Hoojon ,
17/04/2017
Internal Guide

Project seminar was held on 17/04/2017 at Shillong College, Shillong.

Shrimati Aiom Mitri

HOD

Dept. Of Comp. Sci.&App.

Shillong College

Shillong, Meghalaya.

793003

17/4/17
External examinar

Acknowledgement

I would like to take the opportunity to express my deepest thanks to my guide Sir. Ransly hoojon for helping me and encouraging me in doing this project.

I would also like to thank the all staff members of the Department of Computer Science for helping me in every way they can.

Finally, I would Like to thanks the students of our Department, especially Shawn L. Khongwir and Aiboklang Dkhar for their Help, for their Friendship and for encouraging me to do this project.

Thanking You,

MACDONALDKHARSAHNOH

Table of Contents

1. Introduction

A Bible concordanT is an alphabetical listings of words and phrases found in the HOLY BIBLE and shows where the terms occur throughout all books of Scriptures.

2. Synopsis

Project Title: 'KHASI BAIBLE KONKORDANT

Project Technology: Android

Scopes of Project:

To help each individual to easily read and understand the Bible words ,verses and Chapters

3 Working of konkordant

- Perhaps you remember a verse about Noah finding grace in a concordance in order to discover the reference.
- A Bible concordance is useful in locating passages in the Bible. If you can remember just one word in a verse, you can often find what you're looking for.

4 : Existing System study

- The Bible concordance android application is already exist .But the application is only in a foreign language like English,greek,Hebrew.
- What i want to do is that i am going to make this app but the contents and the design will be different.i am going to make this app but it will base on the khasi Bible.And also the user can view all the books and chapters occurs in the Bible and can search for a particular words that he want to know

5. User's Requirements:

Hardware Requirements: Android Mobile Phones,tablets

Software Requirements:

1. OS – Android 4.1 above (Jelly Bean).
2. RAM – 215MB minimum
3. Storage – 13.34MB
4. System Requirements – Single Core above

Front end

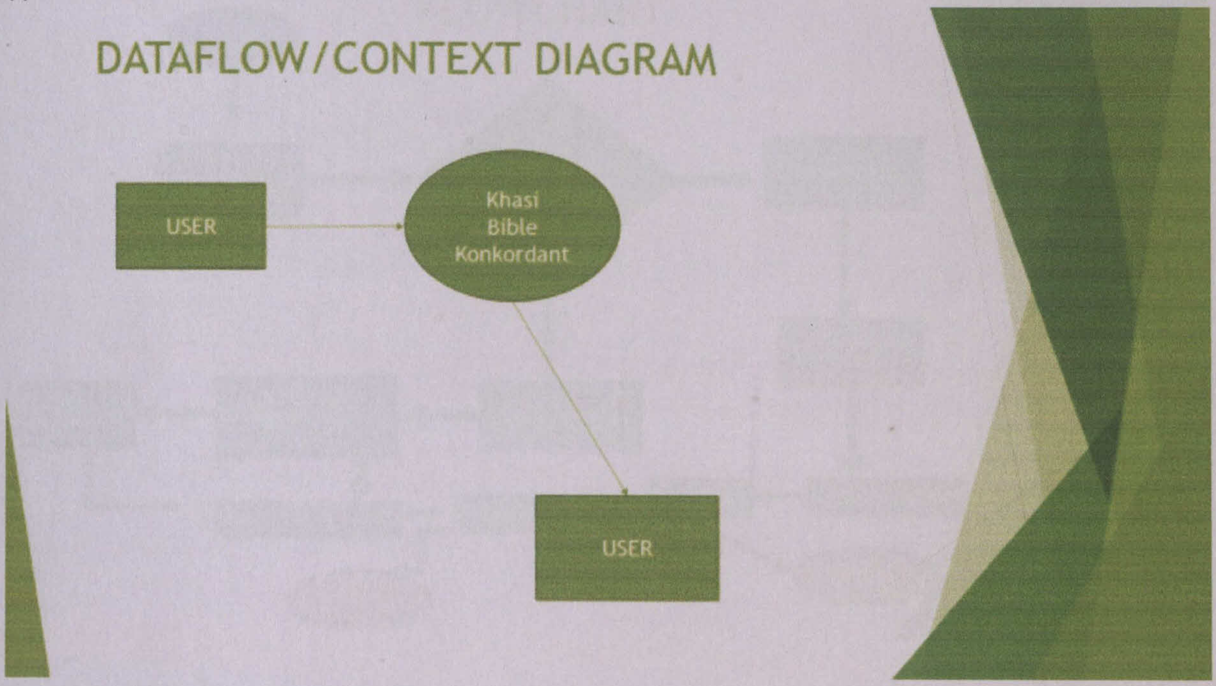
- Front end will consist of xml and java
- Xml is used to design the layout etc.
- Java is used to handle the database and the xml

Back end

- Back end is the database part ,the back end is the part in which we store the file in the database.The back end is handle using the front end.

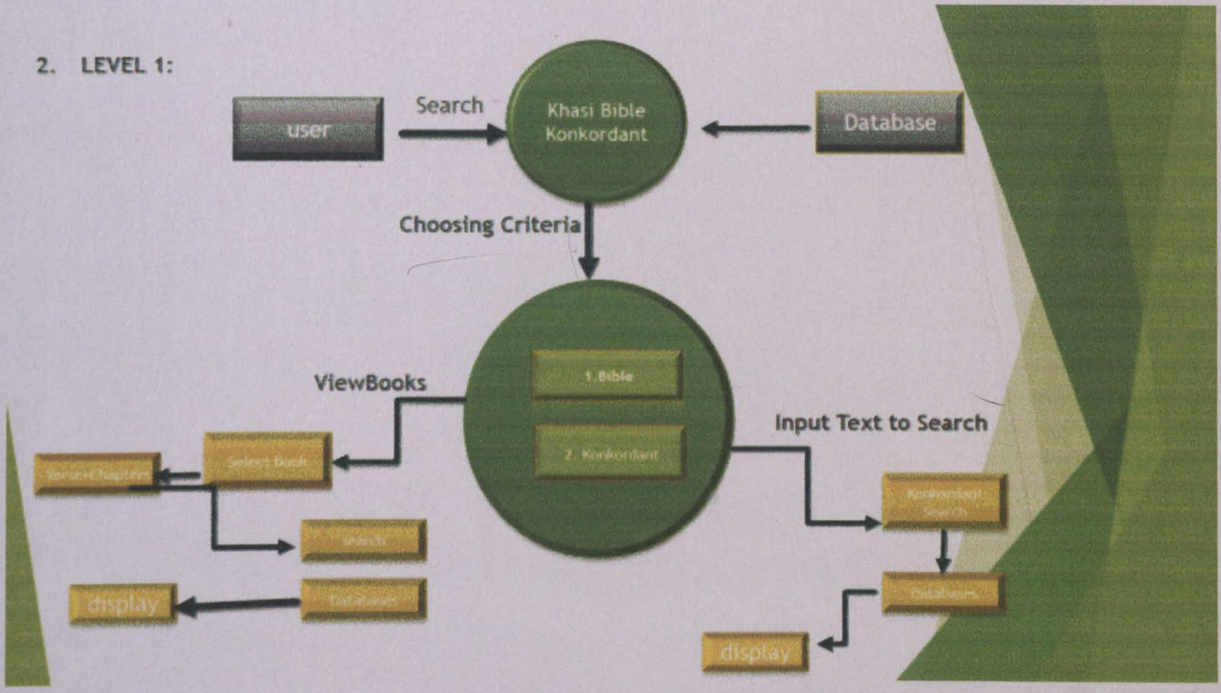
6. DFD Diagram:

(i) Level 0:



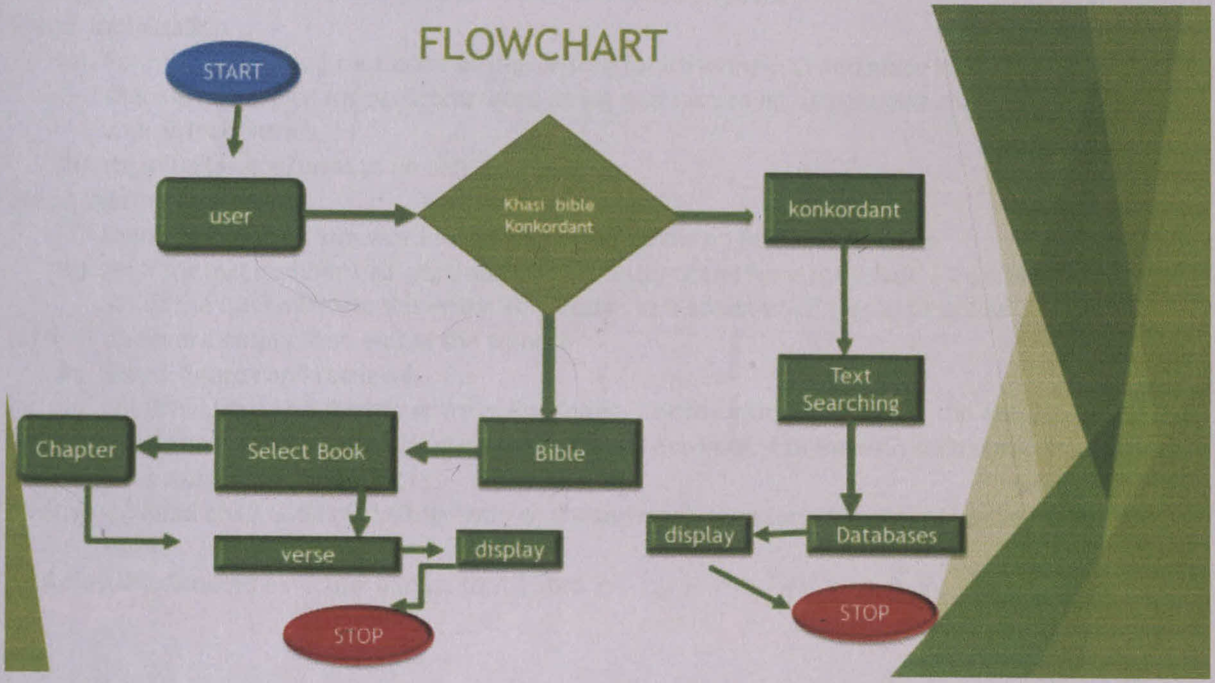
DATAFLOW/CONTEXT DIAGRAM

(ii) Level 1:



2. LEVEL 1:

7:Flowchart:


FLOWCHART

8. Algorithm: INVERTED INDEX ALGORITHM FOR TEXT RETRIEVAL

Step1. Initialization.

(a) For i =1,...,|Q| read root node of the SB-tree for ith word in Q and place its entries in Stack(i). If there is no SB-tree for particular word its list numbers must be decoded and placed in stack as well as leaf entries.

(b) (b) Initialize PosZones to be actual.

Step2. Normalization.

(a) Delete all entries from word stacks which intersects no PosZones extents.

(b) (b) If for text number t all word stacks have entry of the form (t,t,0,NULL) then add t to the result set of the query. Delete this entry from stacks and adjust PosZones to be actual.

(c) If all stacks are empty then end of the algorithm

▶ Step3. Search and retrieval.

(a) Let (tmin,tmax) be the top entry of PosZones . Choose word stack which top entry extent (t0,t1) satisfy t0 <tmin <t 1, and value of dead space is maximal. If there is no such stack choose from ones satisfying t0 ≤ tmin ≤ t1.

(b) (b) Read child node of the top entry of chosen stack and place its entries instead top entry into stack.

(c) Adjust PosZonesto be actual and go to the step 2.

9 Bibliography:

- https://dzone.com/articles/listview-data-sqlitedatabase

- https://developer.android.com/training/basics/data-storage/databases.html
- http://www.androidhive.info/2012/09/android-adding-search-functionality-to-listview/

- http://googleweblight.com/i?u=http://www.c-sharpcorner.com/UploadFile/ef3808/how-to-open-new-activity-on-click-button-by-existing-activi/&grqid=jv9qZJ7S&hl=en-IN

- http://stackoverflow.com/questions/9097723/adding-an-onclicklistener-to-listview-android

XML

SOURCE CODE

Manifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.khasibiblekonkordant"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="23" />
    <application android:icon="@drawable/icon"
        android:label="@string/app_name"
        android:theme="@style/Theme.AppCompat.NoActionBar">
        <activity android:name=".Sdang">

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

        </activity>

        <activity android:name=".Bible"
            android:label="@string/app_name"
            android:theme="@android:style/Theme.Light">

        </activity>

        <activity android:name=".SearchActivity"
            android:label="Search"
            android:theme="@android:style/Theme.Light" />

        <activity android:name=".SearchResultsActivity"
            android:label="Search Results"
            android:theme="@android:style/Theme.Light" />

        <activity android:name=".ChapterActivity"
            android:label="@string/app_name"
            android:theme="@android:style/Theme.Light" />

        <provider android:name=".BibleProvider"
            android:authorities="com.bibleconcordance.bibleprovider" />
    </application>
</manifest>
```

Layout

bible.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/colorPrimary">

     <ListView android:id="@id/android:list"
               android:layout_width="fill_parent"
               android:layout_height="match_parent" />

</LinearLayout>
```

book_row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content">

  <ImageView android:id="@+id/book_image"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:src="@drawable/closed3"
    android:clickable="false"
    android:focusable="false"
    android:paddingLeft="5dip"
    android:paddingRight="5dip" />

  <TextView android:id="@+id/book_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@id/book_image"
    android:layout_centerVertical="true"
    android:text="testing"
    android:textSize="24sp"
    android:padding="5sp"
    android:clickable="false"
    android:focusable="false"
      android:textColor="#ffffff" />

  <ImageView android:id="@+id/book_chapter_selection"
    android:src="@drawable/listing3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_alignParentRight="true"
    android:paddingTop="5sp"
    android:paddingBottom="5sp"
    android:paddingRight="5sp"
    android:clickable="false"
    android:focusable="false" />

</RelativeLayout>
```

Layout

bible.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/colorPrimary">

     <ListView android:id="@id/android:list"
               android:layout_width="fill_parent"
               android:layout_height="match_parent" />

</LinearLayout>
```

book_row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content">

  <ImageView android:id="@+id/book_image"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:src="@drawable/closed3"
    android:clickable="false"
    android:focusable="false"
    android:paddingLeft="5dip"
    android:paddingRight="5dip" />

  <TextView android:id="@+id/book_name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@id/book_image"
    android:layout_centerVertical="true"
    android:text="testing"
    android:textSize="24sp"
    android:padding="5sp"
    android:clickable="false"
    android:focusable="false"
      android:textColor="#ffffff" />

  <ImageView android:id="@+id/book_chapter_selection"
    android:src="@drawable/listing3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_alignParentRight="true"
    android:paddingTop="5sp"
    android:paddingBottom="5sp"
    android:paddingRight="5sp"
    android:clickable="false"
    android:focusable="false" />

</RelativeLayout>
```

change_font.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

    <SeekBar android:id="@+id/change_font_seekbar"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:padding="25sp" />
</LinearLayout>
```

chapter.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="@color/colorPrimary">

  <TextView android:id="@+id/chapter_heading"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content" />

  <ListView android:id="@id/android:list"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent" />

  <ViewStub
      android:id="@+id/stub_navigation"
      android:inflatedId="@+id/panel_navigation"

      android:layout="@layout/navigation_overlay"

      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:layout_gravity="bottom"
      android:layout_marginTop="-51dip" />

</LinearLayout>
```

list_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary">
    <TextView
        android:id="@+id/txtitem"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="10dp"/>

</LinearLayout>
```

navigation_overlay.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"

    android:paddingLeft="10dip"
    android:paddingTop="5dip"
    android:paddingRight="10dip"
    android:paddingBottom="4dip"

    android:background="#99000000" >

    <ImageButton android:id="@+id/navigation_previous"
        android:text="Previous"
        android:src="@drawable/back"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="50" />

    <ImageButton android:id="@+id/navigation_next"
        android:text="Next"
        android:src="@drawable/forward"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="50" />

</LinearLayout>
```

Sdang.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1"
    android:background="@drawable/konkord"
>

    <Button
        android:layout_width="match_parent"
        android:layout_height="92dp"
        android:text="KA BIBLE"
        android:textSize="30dp"
        android:id="@+id/start"
        android:textColor="#ffffff"
        android:background="#7e050505"
        android:layout_marginBottom="103dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:outlineProvider="background" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="92dp"
        android:text="KONKORDANT"
        android:textSize="30dp"
        android:id="@+id/start1"
        android:textColor="#ffffff"
        android:background="#7e050505"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
</RelativeLayout>
```

Search.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
    android:background="@color/colorPrimary">

  <EditText android:id="@+id/search_term"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10sp"
    android:hint="Search term" />

   <Spinner android:id="@+id/search_method"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/search_term"
       android:background="@color/colorPrimary"/>

  <Spinner android:id="@+id/search_scope"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
      android:background="@color/colorPrimary"
      android:layout_alignTop="@+id/search_method" />

  <Button android:id="@+id/submit_search"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Wad"
      android:layout_gravity="bottom"
      android:layout_below="@+id/search_term"
      android:layout_alignParentLeft="true"
      android:layout_alignParentStart="true" />

  <ListView android:id="@+id/search_selected_books_list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
      android:layout_above="@+id/submit_search"
      android:background="@color/colorPrimary"/>

</RelativeLayout>
```

Search_results.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical"
    android:background="@color/colorPrimary">

  <TextView android:id="@+id/search_results_header"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
      android:textColor="#ffffff" />

  <ListView android:id="@id/android:list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" />

</LinearLayout>
```

Search_results_row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

  <TextView android:id="@+id/search_result_row_book"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:textStyle="bold"
    android:text="Song of Solomon"
      android:background="@color/colorPrimary"
      android:textColor="#ffffff" />

  <TextView android:id="@+id/search_result_row_book_chapter_separater"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_toRightOf="@id/search_result_row_book"
    android:textStyle="bold"
    android:text=" - "
      android:textColor="#ffffff" />

  <TextView android:id="@+id/search_result_row_chapter"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_toRightOf="@id/search_result_row_book_chapter_separater"
    android:textStyle="bold"
    android:text="Chapter 1"
      android:textColor="#ffffff" />

  <TextView android:id="@+id/search_result_row_verse_number"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/search_result_row_book"
    android:layout_alignParentLeft="true"
    android:text="[1]"
      android:textColor="#ffffff" />

  <TextView android:id="@+id/search_result_row_verse_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/search_result_row_book"
    android:layout_toRightOf="@id/search_result_row_verse_number"
    android:paddingLeft="5dip"
    android:text="..."
      android:textColor="#ffffff"/>

</RelativeLayout>
```

```
Verse_row.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/colorPrimary">

  <TextView android:id="@+id/verse_number"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="[10]"
      android:textSize="10dp"
      android:textColor="#fefefe" />

  <TextView android:id="@+id/verse_text"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:layout_toRightOf="@id/verse_number"
      android:paddingLeft="5dip"
      android:text="testing"
      android:textSize="14dp"
      android:textColor="#fcfafa" />

</RelativeLayout>
```

Values

```
Strings.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Ka Bible Konkordant</string>
</resources>
```

```
arrays.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string-array name="search_scope">
        <item>All Books</item>
        <item>Old Testament</item>
        <item>New Testament</item>
        <item>Select Books...</item>
    </string-array>

    <string-array name="search_methods">
        <item>Exact phrase</item>
        <item>All words in search term</item>
        <item>Any words in search term</item>
    </string-array>
</resources>
```

```
Colors.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#036772</color>
    <color name="colorPrimaryDark">#303f9f</color>
    <color name="colorAccent">#ff4081</color>
</resources>
```

Styles.xml

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>
```

JAVA

SOURCE CODE

BibleDatabasesHelper.java

```
package com.khasibiblekonkordant;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.List;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.text.SpannableString;
import android.util.Log;
import android.widget.TextView;

public class BibleDatabaseHelper extends SQLiteOpenHelper {

    private static final String TAG = "BibleDatabaseHelper";
    private static final String DATABASE_NAME = "bible.db";
    private SQLiteDatabase database;
    private final Context context;
       private SQLiteOpenHelper openHelper;
       private static BibleDatabaseHelper instance;
       public static final String TEXT = "VerseText";
       public static final String BOOK_ID = "BookID";
       String searchTerm;



       public BibleDatabaseHelper(Context context) {
       super(context, DATABASE_NAME, null, 1);
       this.context = context;
     }


       public static BibleDatabaseHelper getInstance(Context context) {
              if (instance == null) {
                     instance = new BibleDatabaseHelper(context);
              }
              return instance;
        }


     @Override
     public void onCreate(SQLiteDatabase arg0) {
        // TODO Auto-generated method stub

     }

     @Override
     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // TODO Auto-generated method stub

     }

     public String databasePath() {
        return context.getDatabasePath(DATABASE_NAME).getAbsolutePath();
     }
// pyrshang


    public SQLiteDatabase openDatabase() throws SQLException {
       String path = databasePath();
       database = SQLiteDatabase.openDatabase(path, null,
SQLiteDatabase.OPEN_READONLY);
       return database;
    }

    public synchronized void close() {
       if (database != null) {
           database.close();
       }

       super.close();
    }


}
```

```
    public void createDatabase() throws IOException {
      boolean exists = checkDatabase();

      if (!exists) {
          database = this.getReadableDatabase();
          try {
             copyDatabase();
          }
          catch (IOException e) {
             throw new Error("Error copying database");
          }

          if (database != null) {
             database.close();
             database = null;
          }
      }
    }

    public boolean checkDatabase() {
      SQLiteDatabase checkDB = null;

      try {
          String path = databasePath();
          checkDB = SQLiteDatabase.openDatabase(path, null,
SQLiteDatabase.OPEN_READONLY);
      }
      catch (SQLiteException e) {
          // Database doesn't exist
      }

      if (checkDB != null) {
          checkDB.close();
      }

      return (checkDB != null);
    }

    private void copyDatabase() throws IOException {
      String destination = databasePath();
      Log.d(TAG, "opening output stream for database " + destination);
      OutputStream output = new FileOutputStream(destination);

      int[] resources = new int[] { R.raw.bible};
      for (int resource : resources) {
          InputStream input = context.getResources()
                 .openRawResource(resource);

          byte[] buffer = new byte[1024];
          int length;
          while ((length = input.read(buffer)) > 0) {
             output.write(buffer, 0, length);
          }

          output.flush();
          input.close();
      }
      output.close();
    }
```

```
// pyrshang


    public SQLiteDatabase openDatabase() throws SQLException {
       String path = databasePath();
       database = SQLiteDatabase.openDatabase(path, null,
SQLiteDatabase.OPEN_READONLY);
       return database;
    }

    public synchronized void close() {
       if (database != null) {
           database.close();
       }

       super.close();
    }


}
```

BibleSearchTask.java

```
package com.khasibiblekonkordant;

import java.util.List;
import java.util.StringTokenizer;

import android.os.AsyncTask;
import android.util.Log;

public class BibleSearchTask extends AsyncTask<Void, Void, List<Verse>> {

    private static final String TAG = "BibleSearchTask";

    private SearchResultsActivity activity;

    public BibleSearchTask(final SearchResultsActivity activity) {
       this.activity = activity;
    }

    @Override
    protected List<Verse> doInBackground(Void... arg0) {

       StringBuilder builder = new StringBuilder();
             switch (activity.method) {
       case SearchActivity.SEARCH_METHOD_EXACT_PHRASE:

           builder.append("VerseText LIKE '%" + activity.searchTerm + "%'");
           break;
       case SearchActivity.SEARCH_METHOD_ALL_WORDS:

       {
           StringTokenizer tokenizer = new StringTokenizer(activity.searchTerm, " ,.");
           while (tokenizer.hasMoreTokens()) {
              builder.append("VerseText LIKE '%" + tokenizer.nextToken() + "%'");
              if (tokenizer.hasMoreTokens()) {
                  builder.append(" AND ");
              }
           }
       }
           break;
       case SearchActivity.SEARCH_METHOD_ANY_WORDS:

       {
           StringTokenizer tokenizer = new StringTokenizer(activity.searchTerm, " ,.");
           while (tokenizer.hasMoreTokens()) {
              builder.append("VerseText LIKE '%" + tokenizer.nextToken() + "%'");
              if (tokenizer.hasMoreTokens()) {
                  builder.append(" OR ");
              }
           }
       }
           break;
       }


       switch (activity.scope) {
       case SearchActivity.SEARCH_SCOPE_NT_BOOKS:
                  builder.append(" AND BookID IN (SELECT id FROM Books WHERE TestamentID = 2)");
           break;
       case SearchActivity.SEARCH_SCOPE_OT_BOOKS:
           builder.append(" AND BookID IN (SELECT id FROM Books WHERE TestamentID = 1)");
           break;
       case SearchActivity.SEARCH_SCOPE_SELECTED_BOOKS:
                  builder.append(" AND BookID IN (");
           for (int i=0; i<activity.selectedBooks.length; i++) {
              builder.append(activity.selectedBooks[i]);
              if (i < (activity.selectedBooks.length-1)) {
                  builder.append(", ");
              }
```

```
            }
            builder.append(")");
            break;
        }
        Log.d(TAG, "Searching with : " + builder.toString());
        List<Verse> verses = BibleLibrary.getVerses(activity.getContentResolver(),
builder.toString());
        Log.d(TAG, "There were " + verses.size() + " verses found");

        return verses;
    }

    @Override
    protected void onPostExecute(List<Verse> verses) {
        activity.searchCompleted(verses);

         super.onPostExecute(verses);
    }
}
```

BibleProvider.java

```
package com.khasibiblekonkordant;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;


import android.content.ContentProvider;
import android.content.ContentValues;
import android.content.UriMatcher;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteQueryBuilder;
import android.net.Uri;
import android.util.Log;

public class BibleProvider extends ContentProvider {

    private static final String TAG = "BibleProvider";

    private static final int TESTAMENTS = 1;
    private static final int TESTAMENT_ID = 2;
    private static final int BOOKS = 3;
    private static final int BOOK_ID = 4;
    private static final int CHAPTERS = 5;
    private static final int CHAPTER_ID = 6;
    private static final int CHAPTERS_COUNT = 7;
    private static final int VERSES = 8;
    private static final int VERSE_NUMBER = 9;
    private static final int VERSES_COUNT = 10;
    private static final int VERSE_ID = 11;

    public static final String AUTHORITY = "com.bibleconcordance.bibleprovider";

    public static final Uri CONTENT_URI = Uri.parse("content://" + AUTHORITY);



    private static final UriMatcher URI_MATCHER;

    static {
        URI_MATCHER = new UriMatcher(UriMatcher.NO_MATCH);
        URI_MATCHER.addURI(AUTHORITY, "testaments", TESTAMENTS);
        URI_MATCHER.addURI(AUTHORITY, "testaments/#", TESTAMENT_ID);
        URI_MATCHER.addURI(AUTHORITY, "books", BOOKS);
        URI_MATCHER.addURI(AUTHORITY, "books/#", BOOK_ID);
        URI_MATCHER.addURI(AUTHORITY, "books/#/lynnongs", CHAPTERS);
        URI_MATCHER.addURI(AUTHORITY, "books/#/lynnongs/#", CHAPTER_ID);
        URI_MATCHER.addURI(AUTHORITY, "books/#/lynnongs/count", CHAPTERS_COUNT);
        URI_MATCHER.addURI(AUTHORITY, "books/#/lynnongs/#/verses", VERSES);
        URI_MATCHER.addURI(AUTHORITY, "books/#/lynnongs/#/verses/#", VERSE_NUMBER);
        URI_MATCHER.addURI(AUTHORITY, "books/#/lynnongs/#/verses/count", VERSES_COUNT);
        URI_MATCHER.addURI(AUTHORITY, "verses", VERSES);
        URI_MATCHER.addURI(AUTHORITY, "verses/#", VERSE_ID);
    }

    BibleDatabaseHelper dbHelper;

    @Override
    public int delete(Uri uri, String selection, String[] selectionArgs) {
       // TODO Auto-generated method stub
       return 0;
    }

    @Override
    public String getType(Uri uri) {
       int match = URI_MATCHER.match(uri);
       switch (match) {
```

```
    case TESTAMENTS:
        return "vnd.android.cursor.dir/vnd.mac.testament";
    case TESTAMENT_ID:
        return "vnd.android.cursor.item/vnd.mac.testament";
    case BOOKS:
        return "vnd.android.cursor.dir/vnd.mac.book";
    case BOOK_ID:
        return "vnd.android.cursor.item/vnd.mac.book";
    case CHAPTERS:
        return "vnd.android.cursor.dir/vnd.mac.chapter";
    case CHAPTER_ID:
        return "vnd.android.cursor.item/vnd.mac.chapter";
    case VERSES:
        return "vnd.android.cursor.dir/vnd.mac.verse";
    case VERSE_NUMBER:
    case VERSE_ID:
        return "vnd.android.cursor.item/vnd.mac.verse";
    default:
        return null;
    }
  }

  @Override
  public Uri insert(Uri uri, ContentValues values) {
    // TODO Auto-generated method stub
    return null;
  }

  @Override
  public boolean onCreate() {
    dbHelper = new BibleDatabaseHelper(getContext());
    if (dbHelper != null) {
        try {
          dbHelper.createDatabase();
          return true;
        } catch (IOException e) {
          Log.e(TAG, e.getMessage(), e);
          return false;
        }
    }
    return false;
  }

  @Override
  public Cursor query(Uri uri, String[] projection, String selection,
        String[] selectionArgs, String sortOrder) {
    int match = URI_MATCHER.match(uri);
    switch (match) {
    case TESTAMENTS:
    case TESTAMENT_ID:
    {
        SQLiteQueryBuilder builder = new SQLiteQueryBuilder();
        SQLiteDatabase sqlDB = dbHelper.getReadableDatabase();
        builder.setTables("Testaments");
        Cursor c = builder.query(sqlDB, projection, selection, selectionArgs, null, null,
sortOrder);
        c.setNotificationUri(getContext().getContentResolver(), uri);
        return c;
    }
    case BOOK_ID:
    case BOOKS:
    {
        SQLiteQueryBuilder builder = new SQLiteQueryBuilder();
        SQLiteDatabase sqlDB = dbHelper.getReadableDatabase();
        builder.setTables("Books");
        Cursor c = builder.query(sqlDB, projection, selection, selectionArgs, null, null,
sortOrder);
        c.setNotificationUri(getContext().getContentResolver(), uri);
        return c;
    }
```

```
        case CHAPTERS:
        case CHAPTER_ID:
        case CHAPTERS_COUNT:
        {
            SQLiteQueryBuilder builder = new SQLiteQueryBuilder();
            SQLiteDatabase sqlDB = dbHelper.getReadableDatabase();
            builder.setTables("Verses");
            Cursor c = builder.query(sqlDB, projection, selection, selectionArgs, null, null,
sortOrder);
            c.setNotificationUri(getContext().getContentResolver(), uri);
            return c;
        }
        case VERSES:
        case VERSE_NUMBER:
        case VERSE_ID:
        case VERSES_COUNT:
        {
            SQLiteQueryBuilder builder = new SQLiteQueryBuilder();
            SQLiteDatabase sqlDB = dbHelper.getReadableDatabase();
            builder.setTables("Verses");
            Cursor c = builder.query(sqlDB, projection, selection, selectionArgs, null, null,
sortOrder);
            c.setNotificationUri(getContext().getContentResolver(), uri);
            return c;
        }
        }

        return null;
    }

    @Override
    public int update(Uri uri, ContentValues values, String selection,
            String[] selectionArgs) {
        // TODO Auto-generated method stub
        return 0;
    }


}
```

Lynnong.java

```java
package com.khasibiblekonkordant;

import java.util.ArrayList;



import android.net.Uri;
import android.provider.BaseColumns;

public class Lynnong implements BaseColumns {
    public static final String DEFAULT_SORT_ORDER = "Lynnong ASC, Verse ASC";

    public static final String BOOK_ID = "BookID";
    public static final String ID = "Lynnong";

    public Integer id = null;
    public Integer bookId = null;
    public ArrayList<Verse> verses = null;


    public Lynnong(Integer id, Integer bookId)
    {
       super();
       this.id = id;
       this.bookId = bookId;
    }

    public static Uri getContentUri(final Book book)
    {
       return getContentUri(book.id);
    }

    public static Uri getContentUri(final int bookId)
    {
       return Uri.parse(BibleProvider.CONTENT_URI + "/books/" + bookId + "/lynnongs");
    }

    public static Uri getContentUri(final Book book, final int chapter)
    {
       return getContentUri(book.id);
    }

    public static Uri getContentUri(final int bookId, final int chapter)
    {
       return Uri.parse(BibleProvider.CONTENT_URI + "/books/" + bookId + "/lynnongs/" + chapter);
    }

    public static Uri getCountUri(final Book book)
    {
       return getCountUri(book.id);
    }

    public static Uri getCountUri(final int bookId)
    {
       return Uri.parse(BibleProvider.CONTENT_URI + "/books/" + bookId + "/lynnongs/count");
    }

    public static String getWhereClause(final Book book)
    {
       return getWhereClause(book.id);
    }

    public static String getWhereClause(final int bookId)
    {
       return "BookID = " + bookId;
    }
}
```

Testament.java

```
package com.khasibiblekonkordant;

import java.util.ArrayList;
import android.net.Uri;
import android.provider.BaseColumns;

public class Testament implements BaseColumns {
    public static final Uri CONTENT_URI = Uri.parse(BibleProvider.CONTENT_URI + "/testaments");

    public static final String DEFAULT_SORT_ORDER = "id ASC";

    public static final String ID = "id";
    public static final String NAME = "Testament";

    public Integer id = null;
    public String name = null;
    public ArrayList<Book> books = null;


    public Testament(Integer id, String name) {
       super();
       this.id = id;
       this.name = name;
    }
}
```

Verse.java

```
package com.khasibiblekonkordant;

import android.net.Uri;
import android.provider.BaseColumns;

public class Verse implements BaseColumns {
    public static final String DEFAULT_SORT_ORDER = "BookID ASC, Lynnong ASC, Verse ASC";

    public static final String ID = "id";
    public static final String NUMBER = "Verse";
    public static final String TEXT = "VerseText";
    public static final String BOOK_ID = "BookID";
    public static final String CHAPTER = "Lynnong";


    public Integer id = null;
    public Integer number = null;
    public String text = null;
    public Integer bookId = null;
    public Integer chapter = null;


    public Verse(Integer id, Integer number, final String text, final Integer bookId, final Integer
chapter) {
        super();
        this.id = id;
        this.number = number;
        this.text = text;
        this.bookId = bookId;
        this.chapter = chapter;
    }

    public static Uri getContentUri(final Book book, final int chapter) {
        return getContentUri(book.id, chapter);
    }

    public static Uri getContentUri(final int bookId, final int chapter) {
        return Uri.parse(BibleProvider.CONTENT_URI + "/books/" + bookId + "/lynnongs/" + chapter +
"/verses");
    }

    public static Uri getContentUri(final Book book, final int chapter, final int verse) {
        return getContentUri(book.id, chapter, verse);
    }

    public static Uri getContentUri(final int bookId, final int chapter, final int verse) {
        return Uri.parse(BibleProvider.CONTENT_URI + "/books/" + bookId + "/lynnongs/" + chapter +
"/verses/" + verse);
    }

    public static Uri getContentUri() {
        return Uri.parse(BibleProvider.CONTENT_URI + "/verses");
    }

    public static Uri getContentUri(final int verseId) {
        return Uri.parse(BibleProvider.CONTENT_URI + "/verses/" + verseId);
    }

    public static Uri getCountUri(final Book book, final int chapter) {
        return getCountUri(book.id, chapter);
    }

    public static Uri getCountUri(final int bookId, final int chapter) {
        return Uri.parse(BibleProvider.CONTENT_URI + "/books/" + bookId + "/lynnongs/" + chapter +
"/verses/count");
    }
    public static String getWhereClause(final Book book, final int chapter) {
        return getWhereClause(book.id, chapter);
    }

    public static String getWhereClause(final int bookId, final int chapter) {
        return "BookID = " + bookId + " AND Lynnong = " + chapter;
    }

    public static String getWhereClause(final Book book, final int chapter, final int verse) {
        return getWhereClause(book.id, chapter, verse);
    }

    public static String getWhereClause(final int bookId, final int chapter, final int verse) {
        return "BookID = " + bookId + " AND Lynnong = " + chapter + " AND Verse = " + verse;
    }

    public static String getWhereClause(final int verseId) {
        return "id = " + verseId;
    }
}
```

VerseAdapter.java

```
package com.khasibiblekonkordant;

import java.util.ArrayList;
import java.util.List;
import android.content.Context;
import android.content.SharedPreferences;
import android.database.Cursor;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;

public class VerseAdapter extends ArrayAdapter<Verse> {

    private static final String TAG = "VerseAdapter";
    public static final String FONT_PREFERENCE = "VerseAdapter.font";

    private static LayoutInflater inflater = null;
    private float verseFontSize = -1.0f;

    public VerseAdapter(Context context, List<Verse> verses) {
       super(context, R.layout.verse_row, verses);

        inflater = (LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

       SharedPreferences prefs = context.getSharedPreferences("VerseAdapter", 0);
       if (prefs.contains(FONT_PREFERENCE)) {
           verseFontSize = prefs.getFloat(FONT_PREFERENCE, -1.0f);
       }

    }



       public static class ViewHolder{
        public TextView verseNumber;
        public TextView verseText;
    }

    public View getView(final int position, View convertView, ViewGroup parent) {
       View row = convertView;
        ViewHolder holder;

       if (row == null) {
           row = inflater.inflate(R.layout.verse_row, null);
            holder = new ViewHolder();
            holder.verseNumber = (TextView)row.findViewById(R.id.verse_number);
            holder.verseText = (TextView)row.findViewById(R.id.verse_text);
            row.setTag(holder);
       }
        else
            holder = (ViewHolder)row.getTag();

       final Verse entry = (Verse)super.getItem(position);

       Log.d(TAG, "Verse: " + entry.number + " - " + entry.text);
       holder.verseNumber.setText("[" + entry.number + "]");
       holder.verseText.setText(entry.text);

       if (verseFontSize > 0) {
           holder.verseText.setTextSize(verseFontSize);
       }

       return row;
    }
    public float getFontSize() {
       if (verseFontSize > 0) {
           return verseFontSize;
       }
       else {
           View row = inflater.inflate(R.layout.verse_row, null);
           return ((TextView)row.findViewById(R.id.verse_text)).getTextSize();
       }
    }

    public void setFontSize(final float fontSize) {
       this.verseFontSize = fontSize;
    }


}
```

SearchResultsActivity.java

```
package com.khasibiblekonkordant;
import android.app.ListActivity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.TextView;
import android.widget.AdapterView.OnItemClickListener;

import java.util.List;

public class SearchResultsActivity extends ListActivity implements OnItemClickListener {

    private static final String TAG = "SearchResultsActivity";

    public static final String SEARCH_METHOD = "search.method";
    public static final String SEARCH_SCOPE = "search.scope";
    public static final String SEARCH_SCOPE_SELECTED_BOOKS = "search.scope.selected.books";
    public static final String SEARCH_TERM = "search.term";
    public String searchTerm;
    public int method;
    public int scope;
    public int[] selectedBooks;

    private ProgressDialog progress;

    @Override
    public void onCreate(Bundle savedInstanceState) {
       Log.d(TAG, "onCreate");

       super.onCreate(savedInstanceState);
        setContentView(R.layout.search_results);
        getListView().setOnItemClickListener(this);
       this.searchTerm = getIntent().getStringExtra(SEARCH_TERM);
       this.method = getIntent().getIntExtra(SEARCH_METHOD,
       SearchActivity.SEARCH_METHOD_EXACT_PHRASE);
       this.scope = getIntent().getIntExtra(SEARCH_SCOPE, SearchActivity.SEARCH_SCOPE_ALL_BOOKS);
       this.selectedBooks = getIntent().getIntArrayExtra(SEARCH_SCOPE_SELECTED_BOOKS);

       ((TextView) findViewById(R.id.search_results_header)).setText("Search results for \"" +
this.searchTerm + "\"");

       search();
    }

    private void search() {
       progress = ProgressDialog.show(this, null, "Searching", true);

       BibleSearchTask searcher = new BibleSearchTask(this);
       searcher.execute();
    }

    public void searchCompleted(final List<Verse> verses) {

       Log.d(TAG, "There were " + verses.size() + " verses found");
       SearchResultsAdapter adapter = new SearchResultsAdapter(this, verses);
       setListAdapter(adapter);
       if (progress != null)
           progress.dismiss();
    }

    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
       Verse verse = ((SearchResultsAdapter)getListAdapter()).getItem(position);
       Book book = BibleLibrary.getBook(getContentResolver(), verse.bookId);
       Intent intent = new Intent(this, ChapterActivity.class);
       intent.putExtra(ChapterActivity.TITLE, book.name);
```

```
        intent.putExtra(ChapterActivity.BOOK_ID, book.id);
        intent.putExtra(ChapterActivity.CHAPTER, verse.chapter);
        intent.putExtra(ChapterActivity.VERSE, verse.number);
        startActivity(intent);
    }
}
```

SearchResultsAdapter.java

```java
package com.khasibiblekonkordant;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.StringTokenizer;
import android.content.Context;
import android.graphics.Color;
import android.text.SpannableString;
import android.text.style.ForegroundColorSpan;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;
import android.widget.TextView.BufferType;

public class SearchResultsAdapter extends ArrayAdapter<Verse> {

    private static final String TAG = "SearchResultsAdapter";
    private static LayoutInflater inflater = null;
    private SearchResultsActivity activity;
    private HashMap<Integer, String> bookNameLookupMap;
    private List<String> searchTokens;

    public SearchResultsAdapter(SearchResultsActivity context, List<Verse> verses) {
       super(context, R.layout.search_results_row, verses);

        inflater = (LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        this.activity = context;
        populateSearchTokens();
        populateBookNameMap();
    }

    public static class ViewHolder {
       public TextView bookName;
       public TextView chapterNumber;
        public TextView verseNumber;
        public TextView verseText;
    }

    public View getView(final int position, View convertView, ViewGroup parent) {
       View row = convertView;
        ViewHolder holder;

       if (row == null) {
           row = inflater.inflate(R.layout.search_results_row, null);
            holder = new ViewHolder();
            holder.bookName = (TextView)row.findViewById(R.id.search_result_row_book);
            holder.chapterNumber = (TextView)row.findViewById(R.id.search_result_row_chapter);
            holder.verseNumber = (TextView)row.findViewById(R.id.search_result_row_verse_number);
            holder.verseText = (TextView)row.findViewById(R.id.search_result_row_verse_text);
            row.setTag(holder);
       }
        else
            holder = (ViewHolder)row.getTag();

       final Verse entry = (Verse)super.getItem(position);

       holder.bookName.setText(bookNameLookupMap.get(entry.bookId));
       holder.chapterNumber.setText("Lynnong " + entry.chapter);
       holder.verseNumber.setText("[" + entry.number + "]");
       holder.verseText.setText(highlightSearchString(entry.text), BufferType.SPANNABLE);

       return row;
    }
    private SpannableString highlightSearchString(final String verseText) {
       SpannableString formattedString = new SpannableString(verseText);

       for (String token : searchTokens) {
           int index = verseText.toUpperCase().indexOf(token);
           while (index != -1) {
              formattedString.setSpan(new ForegroundColorSpan(Color.RED), index, index +
token.length(), 0);


              index = verseText.toUpperCase().indexOf(token, index+1);
           }
       }

       return formattedString;
    }

    private void populateSearchTokens() {
       searchTokens = new ArrayList<String>();
              if (activity.method == SearchActivity.SEARCH_METHOD_EXACT_PHRASE) {
           searchTokens.add(activity.searchTerm.toUpperCase());
       }
       else {
           StringTokenizer tokenizer = new StringTokenizer(activity.searchTerm, " ,.");
           while (tokenizer.hasMoreTokens()) {
              searchTokens.add(tokenizer.nextToken().toUpperCase());
           }
       }
    }

    private void populateBookNameMap() {
       Log.d(TAG, "populateBookNameMap");

       bookNameLookupMap = new HashMap<Integer, String>();

       List<Book> books = BibleLibrary.getBooks(activity.getContentResolver());
       for (Book book : books) {
           bookNameLookupMap.put(book.id, book.name);
       }
    }
}
```

SearchActivity.java

```
package com.khasibiblekonkordant;

import java.util.ArrayList;
import java.util.List;


import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.util.SparseBooleanArray;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.inputmethod.InputMethodManager;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

public class SearchActivity extends Activity implements OnClickListener
{

    private static final String TAG = "SearchActivity";

    public static final int SEARCH_METHOD_EXACT_PHRASE = 0;
    public static final int SEARCH_METHOD_ALL_WORDS = 1;
    public static final int SEARCH_METHOD_ANY_WORDS = 2;

    public static final int SEARCH_SCOPE_ALL_BOOKS = 0;
    public static final int SEARCH_SCOPE_OT_BOOKS = 1;
    public static final int SEARCH_SCOPE_NT_BOOKS = 2;
    public static final int SEARCH_SCOPE_SELECTED_BOOKS = 3;

    List<Book> books = null;
    EditText searchField;
    ListView selectedBooksListView;
    Spinner searchMethodSpinner;
    Spinner searchScopeSpinner;

    @Override
    public void onCreate(Bundle savedInstanceState) {
       Log.d(TAG, "onCreate");

       super.onCreate(savedInstanceState);

       setContentView(R.layout.search);

       searchField = (EditText) findViewById(R.id.search_term);

        books = BibleLibrary.getBooks(getContentResolver());
        selectedBooksListView = (ListView) findViewById(R.id.search_selected_books_list);

        ArrayAdapter<Book> adapter = new ArrayAdapter<Book>(this,
             android.R.layout.simple_list_item_multiple_choice, books);
        selectedBooksListView.setAdapter(adapter);
        selectedBooksListView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
        selectedBooksListView.setVisibility(View.INVISIBLE);

        searchMethodSpinner = (Spinner) findViewById(R.id.search_method);
        searchScopeSpinner = (Spinner) findViewById(R.id.search_scope);
        ArrayAdapter<CharSequence> methodsAdapter = ArrayAdapter.createFromResource(this,
R.array.search_methods,
                android.R.layout.simple_spinner_item);
        methodsAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        searchMethodSpinner.setAdapter(methodsAdapter);

        ArrayAdapter<CharSequence> scopeAdapter = ArrayAdapter.createFromResource(this,
R.array.search_scope,
                android.R.layout.simple_spinner_item);
        scopeAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        searchScopeSpinner.setAdapter(scopeAdapter);

        searchScopeSpinner.setOnItemSelectedListener(new OnItemSelectedListener() {

          public void onItemSelected(AdapterView<?> parent, View view,
                 int position, long id) {
              selectedBooksListView.setVisibility((SEARCH_SCOPE_SELECTED_BOOKS == position) ?
                   View.VISIBLE :
                   View.INVISIBLE);
          }

          public void onNothingSelected(AdapterView<?> parent) {
              selectedBooksListView.setVisibility(View.INVISIBLE);
          }
       });

        selectedBooksListView.setFastScrollEnabled(true);

        ((Button) findViewById(R.id.submit_search)).setOnClickListener(this);

        searchField.setOnEditorActionListener(new TextView.OnEditorActionListener() {

          public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
             if (event != null && (event.getKeyCode() == KeyEvent.KEYCODE_ENTER)) {
                 InputMethodManager in = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
                 in.hideSoftInputFromWindow(searchField.getWindowToken(),
InputMethodManager.HIDE_NOT_ALWAYS);
                 return true;
             }
             return false;
          }
       });
    }

    public void onClick(View v) {

       if ("".equals(searchField.getText().toString())) {
           Toast.makeText(this, "Please enter a search term", Toast.LENGTH_SHORT).show();

       }
       else if (SEARCH_SCOPE_SELECTED_BOOKS == searchScopeSpinner.getSelectedItemPosition() &&
             getSelectedBooks().length < 1) {
           Toast.makeText(this, "No books selected", Toast.LENGTH_SHORT).show();
       }
       else {
           Intent intent = new Intent(this, SearchResultsActivity.class);
           intent.putExtra(SearchResultsActivity.SEARCH_TERM, searchField.getText().toString());
           intent.putExtra(SearchResultsActivity.SEARCH_METHOD,
searchMethodSpinner.getSelectedItemPosition());
           intent.putExtra(SearchResultsActivity.SEARCH_SCOPE,
searchScopeSpinner.getSelectedItemPosition());

           if (SEARCH_SCOPE_SELECTED_BOOKS == searchScopeSpinner.getSelectedItemPosition()) {
             intent.putExtra(SearchResultsActivity.SEARCH_SCOPE_SELECTED_BOOKS,
getSelectedBooks());
           }
           startActivity(intent);
       }
    }
    @SuppressWarnings("unchecked")
    public int[] getSelectedBooks() {
       ArrayList<Integer> selectedBooks = new ArrayList<Integer>();
       SparseBooleanArray items = selectedBooksListView.getCheckedItemPositions();

       ArrayAdapter<Book> adapter = (ArrayAdapter<Book>)selectedBooksListView.getAdapter();
       for (int i=0; i<items.size(); i++) {
           if (items.valueAt(i)) {
             selectedBooks.add(adapter.getItem(items.keyAt(i)).id);
           }
       }

       int[] asIntArray = new int[selectedBooks.size()];

       for (int i=0; i<selectedBooks.size(); i++) {
           asIntArray[i] = selectedBooks.get(i);
       }

       return asIntArray;
    }
}
```

ChapterActivity.java

```
package com.khasibiblekonkordant;

import java.util.List;
import android.app.AlertDialog;
import android.app.ListActivity;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewStub;
import android.view.animation.AnimationUtils;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ImageButton;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;

public class ChapterActivity extends ListActivity implements OnItemLongClickListener {

    private static final String TAG = "ChapterActivity";
    public static final String TITLE = "ChapterActivity.title";
    public static final String BOOK_ID = "ChapterActivity.book_id";
    public static final String CHAPTER = "ChapterActivity.chapter";
    public static final String VERSE = "ChapterActivity.verse";
    private String book;
    private int bookId;
    private int chapter;
    private int verse;
    private List<Verse> verses;
    private View navigationPanel;
    private Handler closeNavigationHandler;
    private Thread closeNavigationThread;

 @Override
    public void onCreate(Bundle savedInstanceState) {
      Log.d(TAG, "onCreate");

      super.onCreate(savedInstanceState);

      setContentView(R.layout.chapter);

      this.book = getIntent().getStringExtra(TITLE);
      this.bookId = getIntent().getIntExtra(BOOK_ID, 1);
      this.chapter = getIntent().getIntExtra(CHAPTER, 1);
      this.verse = getIntent().getIntExtra(VERSE, 0);

      closeNavigationHandler = new Handler();
      closeNavigationThread = new Thread(new Runnable() {

          public void run() {

      navigationPanel.startAnimation(AnimationUtils.loadAnimation(ChapterActivity.this,
                  R.anim.slide_out));
```

```
            navigationPanel.setVisibility(View.GONE);
          }
      });

      loadChapter();

      getListView().setOnItemLongClickListener(this);

      getListView().setOnTouchListener(new View.OnTouchListener() {

          public boolean onTouch(View v, MotionEvent event) {
            if (event.getAction() == MotionEvent.ACTION_DOWN) {
                  if (navigationPanel == null) {
                  navigationPanel = ((ViewStub)
findViewById(R.id.stub_navigation)).inflate();
                  navigationPanel.setVisibility(View.GONE);
                  ImageButton previous =
(ImageButton)navigationPanel.findViewById(R.id.navigation_previous);
                  previous.setBackgroundDrawable(null);
                  previous.setOnClickListener(new View.OnClickListener() {

                      public void onClick(View v) {
                        postHideNavigation();
                        backward();
                      }
                  });

                  ImageButton next =
(ImageButton)navigationPanel.findViewById(R.id.navigation_next);
                  next.setBackgroundDrawable(null);
                  next.setOnClickListener(new View.OnClickListener() {

                      public void onClick(View v) {
                        postHideNavigation();
                        forward();
                      }
                  });
                  }

                  if (navigationPanel.getVisibility() != View.VISIBLE) {

      navigationPanel.startAnimation(AnimationUtils.loadAnimation(ChapterActivity.this,
                        R.anim.slide_in));
                  navigationPanel.setVisibility(View.VISIBLE);
                  }
                  postHideNavigation();
            }
            return false;
          }
      });
    }

    private void postHideNavigation() {
      closeNavigationHandler.removeCallbacks(closeNavigationThread);
      closeNavigationHandler.postDelayed(closeNavigationThread, 3000);
    }

    public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long
id) {
      final Verse selectedVerse = verses.get(position);

      AlertDialog.Builder builder = new AlertDialog.Builder(this);
      builder.setTitle("Thep iaka Bookmark");
      builder.setMessage("Thep iaka bookmark naka bynta " + this.book + " Lynnong " +
selectedVerse.chapter + " verse " + (position + 1) + "?");
      builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int whichButton) {
            Bookmarks bookmarks = new Bookmarks(ChapterActivity.this);
            bookmarks.addBookmark(selectedVerse.id);

            dialog.cancel();
          }
      });
      builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int whichButton) {
      dialog.cancel();
          }
      });

      builder.show();

      return true;
    }

    private void backward() {
      if (this.bookId == 1 && this.chapter == 1) {
          Toast.makeText(this, "Phi don haka basdang jong ka Bible",
Toast.LENGTH_SHORT).show();
      }
      else if (this.chapter > 1) {
                this.chapter--;
          loadChapter();
      }
      else {
          this.bookId--;
          final Book previousBook = BibleLibrary.getBook(getContentResolver(),
this.bookId);
          if (previousBook != null) {
            this.book = previousBook.name;
          }
          int chapterCount = BibleLibrary.getChapterCount(getContentResolver(),
previousBook);
          this.chapter = chapterCount;
          loadChapter();
      }
    }

    private void forward() {

book
      if (this.bookId == 66 && this.chapter == 22) {
          Toast.makeText(this, "Phi la poi sha ka bakut jong ka Bible",
Toast.LENGTH_SHORT).show();

          return;
      }


      int chapterCount = BibleLibrary.getChapterCount(getContentResolver(), this.bookId);

      if (this.chapter == chapterCount) {
                this.bookId++;
          this.chapter = 1;
          final Book nextBook = BibleLibrary.getBook(getContentResolver(), this.bookId);
          if (nextBook != null) {
            this.book = nextBook.name;
          }

          loadChapter();
      }
      else {
          this.chapter++;
          loadChapter();
      }
    }

    public void loadChapter() {
            if (this.book != null)
                  this.setTitle(this.book);

            ((TextView)findViewById(R.id.chapter_heading)).setText("Lynnong " +
this.chapter);

            this.verses = BibleLibrary.getVerses(getContentResolver(), this.bookId,
this.chapter);
            Log.d(TAG, "Loaded " + this.verses.size() + " verses");
            VerseAdapter adapter = new VerseAdapter(this, this.verses);
            setListAdapter(adapter);
            if (this.verse != 0)
                  getListView().setSelection(this.verse-1); // setSelection is 0 based,
verse number is 1 based

            String toast = (this.book != null) ? (this.book + " Lynnong " + this.chapter)
: ("Lynnong " + this.chapter);
            Toast.makeText(this, toast, Toast.LENGTH_SHORT).show();
      }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
      MenuInflater inflater = getMenuInflater();
      inflater.inflate(R.menu.chapter_menu, menu);
      return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
      switch (item.getItemId()) {
      case R.id.load_bookmarks_menu_item:
          loadBookmark();
          break;
      case R.id.change_font_menu_item:
          changeFont();
          break;
      case R.id.select_chapter_menu_item:
          selectChapter();
          break;
      }
      return super.onOptionsItemSelected(item);
    }

    private void loadBookmark() {
      Bookmarks bookmarks = new Bookmarks(this);
      bookmarks.loadBookmarks();
      final List<Integer> bookmarkListing = bookmarks.bookmarks;

      final String[] bookmarkStrings = new String[bookmarkListing.size()];
      if (bookmarkListing.size() == 0) {
          Toast.makeText(this, "Ym don bookmarks Ba phi saved",
Toast.LENGTH_LONG).show();
```

```
      }
      else {
          for (int i=0; i<bookmarkListing.size(); i++) {
            Verse bookmarkedVerse = BibleLibrary.getVerse(getContentResolver(),
bookmarkListing.get(i));
            final Book book = BibleLibrary.getBook(getContentResolver(),
bookmarkedVerse.bookId);
            bookmarkStrings[i] = book.name + " Lynnong " + bookmarkedVerse.chapter + "
Verse " + bookmarkedVerse.number;
          }

          AlertDialog.Builder builder = new AlertDialog.Builder(this);
          builder.setTitle("Load Bookmark");
          builder.setSingleChoiceItems(bookmarkStrings, -1, new
DialogInterface.OnClickListener() {

            public void onClick(DialogInterface dialog, int which) {
                Verse selectedBookmark = BibleLibrary.getVerse(getContentResolver(),
bookmarkListing.get(which));
                final Book book = BibleLibrary.getBook(getContentResolver(),
selectedBookmark.bookId);

                Intent intent = new Intent(ChapterActivity.this, ChapterActivity.class);
                intent.putExtra(ChapterActivity.TITLE, book.name);
                intent.putExtra(ChapterActivity.BOOK_ID, book.id);
                intent.putExtra(ChapterActivity.CHAPTER, selectedBookmark.chapter);
                intent.putExtra(ChapterActivity.VERSE, selectedBookmark.number);
                startActivity(intent);
                dialog.cancel();
            }
          });

          builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int whichButton) {
                dialog.cancel();
            }
          });

          builder.show();
      }
    }

    public void changeFont() {

      final float MIN_FONT_SIZE = 14;
      final float MAX_FONT_SIZE = 26;
      final VerseAdapter adapter = (VerseAdapter)getListAdapter();
      final float originalFontSize = adapter.getFontSize();

      AlertDialog.Builder builder = new AlertDialog.Builder(this);
      builder.setTitle("Pynheh ne Pynrit iaki Dak ");
      View view = getLayoutInflater().inflate(R.layout.change_font, null);
      final SeekBar sizerBar = (SeekBar) view.findViewById(R.id.change_font_seekbar);
      sizerBar.setMax((int)(MAX_FONT_SIZE - MIN_FONT_SIZE));
      sizerBar.setProgress((int)(originalFontSize - MIN_FONT_SIZE));
      sizerBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {

          public void onStopTrackingTouch(SeekBar seekBar) {
            // TODO Auto-generated method stub

          }

          public void onStartTrackingTouch(SeekBar seekBar) {
            // TODO Auto-generated method stub

          }

          public void onProgressChanged(SeekBar seekBar, int progress,
                boolean fromUser) {
            adapter.setFontSize(progress + MIN_FONT_SIZE);
            adapter.notifyDataSetChanged();
          }
      });
      builder.setView(view);
      builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {

          public void onClick(DialogInterface dialog, int which) {
            SharedPreferences prefs =
ChapterActivity.this.getSharedPreferences("VerseAdapter", 0);
            SharedPreferences.Editor editor = prefs.edit();
            editor.putFloat(VerseAdapter.FONT_PREFERENCE, sizerBar.getProgress() +
MIN_FONT_SIZE);
            editor.commit();
          }
      });

      builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

          public void onClick(DialogInterface dialog, int which) {
            adapter.setFontSize(originalFontSize);
            adapter.notifyDataSetChanged();
          }
      });

      builder.show();
    }

    public void selectChapter() {
      final Book book = BibleLibrary.getBook(getContentResolver(), this.bookId);
      int count = BibleLibrary.getChapterCount(getContentResolver(), book);

      final String[] chapterNames = new String[count];
      for (int i=0; i<count; i++) {
          chapterNames[i] = "Lynnong " + (i+1);
      }

      AlertDialog.Builder builder = new AlertDialog.Builder(this);
      builder.setTitle(book.name);
      builder.setSingleChoiceItems(chapterNames, -1, new
DialogInterface.OnClickListener() {

          public void onClick(DialogInterface dialog, int which) {
            ChapterActivity.this.chapter = which+1;
            loadChapter();

            dialog.cancel();
          }
      });

      builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog, int whichButton) {
            dialog.cancel();
          }
      });

      builder.show();
```

Bookmarks.java

```
package com.khasibiblekonkordant;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

import android.content.Context;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;

public class Bookmarks {

    public List<Integer> bookmarks = new ArrayList<Integer>();
    private Context context;

    public Bookmarks(Context context) {
       this.context = context;
    }

    public void loadBookmarks() {
             bookmarks.clear();

       SharedPreferences prefs = context.getSharedPreferences("Bookmarks", 0);
       int ctr = 0;
       String key = "bookmark" + ctr;
       while (prefs.contains(key)) {
           int verseId = prefs.getInt(key, 0);
           if (verseId != 0) {
             bookmarks.add(verseId);
           }

           ctr++;
           key = "bookmark" + ctr;
       }

       Collections.sort(bookmarks);
    }

    public void saveBookmarks() {
       SharedPreferences prefs = context.getSharedPreferences("Bookmarks", 0);
       Editor editor = prefs.edit();
       editor.clear();

       for (int i=0; i<bookmarks.size(); i++) {
           editor.putInt("bookmark" + i, bookmarks.get(i));
       }
       editor.commit();
    }

    public void addBookmark(final Integer verseId) {
       loadBookmarks();

       if (!bookmarks.contains(verseId)) {
           bookmarks.add(verseId);
           Collections.sort(bookmarks);
           saveBookmarks();
       }
    }

    public void removeBookmark(final Integer verseId) {
       loadBookmarks();

       if (bookmarks.contains(verseId)) {
           bookmarks.remove(verseId);
           Collections.sort(bookmarks);
           saveBookmarks();
       }
    }
```

Bookmarks.java

```
package com.khasibiblekonkordant;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

import android.content.Context;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;

public class Bookmarks {

    public List<Integer> bookmarks = new ArrayList<Integer>();
    private Context context;

    public Bookmarks(Context context) {
       this.context = context;
    }

    public void loadBookmarks() {
             bookmarks.clear();

       SharedPreferences prefs = context.getSharedPreferences("Bookmarks", 0);
       int ctr = 0;
       String key = "bookmark" + ctr;
       while (prefs.contains(key)) {
           int verseId = prefs.getInt(key, 0);
           if (verseId != 0) {
              bookmarks.add(verseId);
           }

           ctr++;
           key = "bookmark" + ctr;
       }

       Collections.sort(bookmarks);
    }

    public void saveBookmarks() {
       SharedPreferences prefs = context.getSharedPreferences("Bookmarks", 0);
       Editor editor = prefs.edit();
       editor.clear();

       for (int i=0; i<bookmarks.size(); i++) {
           editor.putInt("bookmark" + i, bookmarks.get(i));
       }
       editor.commit();
    }

    public void addBookmark(final Integer verseId) {
       loadBookmarks();

       if (!bookmarks.contains(verseId)) {
           bookmarks.add(verseId);
           Collections.sort(bookmarks);
           saveBookmarks();
       }
    }

    public void removeBookmark(final Integer verseId) {
       loadBookmarks();

       if (bookmarks.contains(verseId)) {
           bookmarks.remove(verseId);
           Collections.sort(bookmarks);
           saveBookmarks();
       }
    }
}
```

BibleLibrary.java

```
package com.khasibiblekonkordant;

import java.util.ArrayList;
import java.util.List;
import android.content.ContentResolver;
import android.database.Cursor;
import android.util.Log;

public class BibleLibrary {
    private static final String TAG = "BibleLibrary";

    public static List<Testament> getTestaments(final ContentResolver resolver) {
       List<Testament> testaments = new ArrayList<Testament>();
       Cursor cursor = null;
       try {
           try {
              cursor = getTestamentsCursor(resolver);

              while (cursor.moveToNext()) {
                  testaments.add(new Testament(cursor.getInt(cursor.getColumnIndex(Testament.ID)),
                         cursor.getString(cursor.getColumnIndex(Testament.NAME))));
              }
           } finally {
              if (cursor != null) {
                  cursor.close();
              }
           }
       }
       catch (Exception e) {
           Log.e(TAG, e.getMessage(), e);
       }
       return testaments;
    }

    public static Cursor getTestamentsCursor(final ContentResolver resolver) {
       return resolver.query(Testament.CONTENT_URI, new String[] { Testament.ID, Testament.NAME },
              null, null, Testament.DEFAULT_SORT_ORDER);
    }

    public static List<Book> getBooks(final ContentResolver resolver) {
       List<Book> books = new ArrayList<Book>();

       Cursor cursor = null;
       try {
           try {
              cursor = getBooksCursor(resolver);

              while (cursor.moveToNext()) {
                  books.add(new Book(cursor.getInt(cursor.getColumnIndex(Book.ID)),
cursor.getString(cursor.getColumnIndex(Book.NAME))));
              }
           } finally {
              if (cursor != null) {
                  cursor.close();
              }
           }
       } catch (Exception e) {
           Log.e(TAG, e.getMessage(), e);
       }

       return books;
    }

    public static Cursor getBooksCursor(final ContentResolver resolver) {
       return resolver.query(Book.CONTENT_URI, new String[] { Book.ID, Book.NAME },
              null, null, Book.DEFAULT_SORT_ORDER);
    }
    public static List<Book> getBooks(final ContentResolver resolver, final Testament testament) {
       return getBooks(resolver, testament.id);
    }

    public static List<Book> getBooks(final ContentResolver resolver, final int testamentId) {
       List<Book> books = new ArrayList<Book>();

       Cursor cursor = null;
       try {
           try {
              cursor = getBooksCursor(resolver, testamentId);

              while (cursor.moveToNext()) {
                  books.add(new Book(cursor.getInt(cursor.getColumnIndex(Book.ID)),
cursor.getString(cursor.getColumnIndex(Book.NAME))));
              }
           } finally {
              if (cursor != null) {
                  cursor.close();
              }
           }
       } catch (Exception e) {
           Log.e(TAG, e.getMessage(), e);
       }

       return books;
    }

    public static Cursor getBooksCursor(final ContentResolver resolver, final Testament testament)
{
       return getBooksCursor(resolver, testament.id);
    }

    public static Cursor getBooksCursor(final ContentResolver resolver, final int testamentId) {
       return resolver.query(Book.CONTENT_URI, new String[] { Book.ID, Book.NAME },
              "TestamentID = " + testamentId, null, Book.DEFAULT_SORT_ORDER);
    }

    public static Book getBook(final ContentResolver resolver, final int bookId) {
       Book book = null;

       Cursor cursor = null;
       try {
           try {
              cursor = resolver.query(Book.getContentUri(bookId),
                     new String[] { Book.ID, Book.NAME },
                     Book.getWhereClause(bookId), null,
                     Book.DEFAULT_SORT_ORDER);

              while (cursor.moveToNext()) {
                  book = new Book(cursor.getInt(cursor.getColumnIndex(Book.ID)),
                         cursor.getString(cursor.getColumnIndex(Book.NAME)));
              }
           } finally {
              if (cursor != null) {
                  cursor.close();
              }
           }
       } catch (Exception e) {
           Log.e(TAG, e.getMessage(), e);
       }
       return book;
    }


    public static int getChapterCount(final ContentResolver resolver, final Book book) {
       return getChapterCount(resolver, book.id);
    }
    public static int getChapterCount(final ContentResolver resolver, final int bookId) {
       int count = 0;

       Cursor cursor = null;
       try {
           try {
              cursor = resolver.query(Lynnong.getCountUri(bookId), new String[] { "MAX(Lynnong) AS 
count" },
                     Lynnong.getWhereClause(bookId), null, "count");

              while (cursor.moveToNext()) {
                  count = cursor.getInt(cursor.getColumnIndex("count"));
              }
           } finally {
              if (cursor != null) {
                  cursor.close();
              }
           }
       } catch (Exception e) {
           Log.e(TAG, e.getMessage(), e);
       }

       return count;
    }

    public static List<Lynnong> getChapters(final ContentResolver resolver, final Book book) {
       return getChapters(resolver, book.id);
    }

    public static List<Lynnong> getChapters(final ContentResolver resolver, final int bookId) {
       List<Lynnong> lynnongs = new ArrayList<Lynnong>();

       Cursor cursor = null;
       try {
           try {
              cursor = getChaptersCursor(resolver, bookId);

              while (cursor.moveToNext()) {
                  lynnongs.add(new Lynnong(cursor.getInt(cursor.getColumnIndex(Lynnong.ID)), 
cursor.getInt(cursor.getColumnIndex(Lynnong.BOOK_ID))));
              }
           } finally {
              if (cursor != null) {
                  cursor.close();
              }
           }
       } catch (Exception e) {
           Log.e(TAG, e.getMessage(), e);
       }

       return lynnongs;
    }

    public static Cursor getChaptersCursor(final ContentResolver resolver, final Book book) {
       return getChaptersCursor(resolver, book.id);
    }

    public static Cursor getChaptersCursor(final ContentResolver resolver, final int bookId) {
       return resolver.query(Lynnong.getContentUri(bookId), new String[] { Lynnong.ID, 
Lynnong.BOOK_ID },
              Lynnong.getWhereClause(bookId), null, Lynnong.DEFAULT_SORT_ORDER);
    }

    // Get one chapter in a book
    public static Lynnong getChapter(final ContentResolver resolver, final Book book, final int 
chapter) {
       return getChapter(resolver, book.id, chapter);
    }
    public static Lynnong getChapter(final ContentResolver resolver, final int bookId, final int
chapter) {
        return null;
    }

    public static int getVerseCount(final ContentResolver resolver, final Book book, final int
chapter) {
        return getVerseCount(resolver, book.id, chapter);
    }

    public static int getVerseCount(final ContentResolver resolver, final int bookId, final int
chapter) {
        int count = 0;

        Cursor cursor = null;
        try {
            try {
                cursor = resolver.query(Verse.getCountUri(bookId, chapter), new String[] { "MAX(Verse)
AS count" },
                        Verse.getWhereClause(bookId, chapter), null, "count");

                while (cursor.moveToNext()) {
                    count = cursor.getInt(cursor.getColumnIndex("count"));
                }
            } finally {
                if (cursor != null) {
                    cursor.close();
                }
            }
        } catch (Exception e) {
            Log.e(TAG, e.getMessage(), e);
        }

        return count;
    }

    public static List<Verse> getVerses(final ContentResolver resolver, final Book book, final int
chapter) {
        return getVerses(resolver, book.id, chapter);
    }

    public static List<Verse> getVerses(final ContentResolver resolver, final int bookId, final int
chapter) {
        List<Verse> verses = new ArrayList<Verse>();

        Cursor cursor = null;
        try {
            try {
                cursor = getVersesCursor(resolver, bookId, chapter);

                while (cursor.moveToNext()) {
                    verses.add(new Verse(cursor.getInt(cursor.getColumnIndex(Verse.ID)),
                            cursor.getInt(cursor.getColumnIndex(Verse.NUMBER)),
                            cursor.getString(cursor.getColumnIndex(Verse.TEXT)),
                            cursor.getInt(cursor.getColumnIndex(Verse.BOOK_ID)),
                            cursor.getInt(cursor.getColumnIndex(Verse.CHAPTER))));
                }
            } finally {
                if (cursor != null) {
                    cursor.close();
                }
            }
        } catch (Exception e) {
            Log.e(TAG, e.getMessage(), e);
        }

        return verses;
    }
    public static Cursor getVersesCursor(final ContentResolver resolver, final Book book, final int
chapter) {
        return getVersesCursor(resolver, book.id, chapter);
    }

    public static Cursor getVersesCursor(final ContentResolver resolver, final int bookId, final
int chapter) {
        return resolver.query(Verse.getContentUri(bookId, chapter), new String[] { Verse.ID,
Verse.NUMBER, Verse.TEXT, Verse.BOOK_ID, Verse.CHAPTER },
                Verse.getWhereClause(bookId, chapter), null, Verse.DEFAULT_SORT_ORDER);
    }

    public static Verse getVerse(final ContentResolver resolver, final Book book, final int
chapter, final int verse) {
        return getVerse(resolver, book.id, chapter, verse);
    }

    public static Verse getVerse(final ContentResolver resolver, final int bookId, final int
chapter, final int verse) {
        Verse v = null;

        Cursor cursor = null;
        try {
            try {
                cursor = resolver.query(Verse.getContentUri(bookId, chapter, verse),
                        new String[] { Verse.ID, Verse.NUMBER, Verse.TEXT, Verse.BOOK_ID, Verse.CHAPTER
},

                        Verse.getWhereClause(bookId, chapter, verse), null,
                        Verse.DEFAULT_SORT_ORDER);


                while (cursor.moveToNext()) {
                    v = new Verse(cursor.getInt(cursor.getColumnIndex(Verse.ID)),
                            cursor.getInt(cursor.getColumnIndex(Verse.NUMBER)),
                            cursor.getString(cursor.getColumnIndex(Verse.TEXT)),
                            cursor.getInt(cursor.getColumnIndex(Verse.BOOK_ID)),
                            cursor.getInt(cursor.getColumnIndex(Verse.CHAPTER)));
                }
            } finally {
                if (cursor != null) {
                    cursor.close();
                }
            }
        } catch (Exception e) {
            Log.e(TAG, e.getMessage(), e);
        }
        return v;
    }

    public static Verse getVerse(final ContentResolver resolver, final int verseId) {
        Verse v = null;

        Cursor cursor = null;
        try {
            try {
                cursor = resolver.query(Verse.getContentUri(verseId),
                        new String[] { Verse.ID, Verse.NUMBER, Verse.TEXT, Verse.BOOK_ID, Verse.CHAPTER
},

                        Verse.getWhereClause(verseId), null, Verse.DEFAULT_SORT_ORDER);

                while (cursor.moveToNext()) {
                    v = new Verse(cursor.getInt(cursor.getColumnIndex(Verse.ID)),
                            cursor.getInt(cursor.getColumnIndex(Verse.NUMBER)),
                            cursor.getString(cursor.getColumnIndex(Verse.TEXT)),
                            cursor.getInt(cursor.getColumnIndex(Verse.BOOK_ID)),
                            cursor.getInt(cursor.getColumnIndex(Verse.CHAPTER)));
                }
            } finally {
                if (cursor != null) {
                    cursor.close();
            }
        }
    } catch (Exception e) {
        Log.e(TAG, e.getMessage(), e);
    }
    return v;
  }

  public static List<Verse> getVerses(final ContentResolver resolver, final String where) {
    List<Verse> verses = new ArrayList<Verse>();

    Cursor cursor = null;
    try {
        try {
          cursor = getVersesCursor(resolver, where);

          while (cursor.moveToNext()) {
              verses.add(new Verse(cursor.getInt(cursor.getColumnIndex(Verse.ID)),
                      cursor.getInt(cursor.getColumnIndex(Verse.NUMBER)),
                      cursor.getString(cursor.getColumnIndex(Verse.TEXT)),
                      cursor.getInt(cursor.getColumnIndex(Verse.BOOK_ID)),
                      cursor.getInt(cursor.getColumnIndex(Verse.CHAPTER))));
          }
        } finally {
          if (cursor != null) {
              cursor.close();
          }
        }
    } catch (Exception e) {
        Log.e(TAG, e.getMessage(), e);
    }

    return verses;
  }


  public static Cursor getVersesCursor(final ContentResolver resolver, final String where) {
    return resolver.query(Verse.getContentUri(), new String[] { Verse.ID, Verse.NUMBER,
Verse.TEXT, Verse.BOOK_ID, Verse.CHAPTER },
              where, null, Verse.DEFAULT_SORT_ORDER);
  }
}
```

Sdang.java

```
package com.khasibiblekonkordant;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;

public class Sdang extends AppCompatActivity
{
    public Button start,start1;

    public void initmenu()
    {
        start =(Button)findViewById(R.id.start);
        start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent start= new Intent(Sdang.this,Bible.class);
                startActivity(start);
            }
        });
    }

    public void concordance()
    {
        start1 =(Button)findViewById(R.id.start1);
        start1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent start1= new Intent(Sdang.this,SearchActivity.class);
                startActivity(start1);
            }
        });
    }


    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sdang);

        initmenu();
        concordance();


    }


}
```

BookAdapter.java

```
package com.khasibiblekonkordant;

import java.util.List;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

public class BookAdapter extends ArrayAdapter<Book> {

    private static LayoutInflater inflater = null;
    private Bible context;

    public BookAdapter(Bible context, List<Book> books) {
       super(context, R.layout.book_row, books);

       this.context = context;
        inflater = (LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    public static class ViewHolder{
        public TextView bookName;
        public ImageView chapterSelectButton;
    }

    public View getView(final int position, View convertView, ViewGroup parent) {
       View row = convertView;
        ViewHolder holder;

       if (row == null) {
           row = inflater.inflate(R.layout.book_row, null);
            holder = new ViewHolder();
            holder.bookName = (TextView)row.findViewById(R.id.book_name);
            holder.chapterSelectButton = (ImageView)row.findViewById(R.id.book_chapter_selection);
            row.setTag(holder);
       }
        else
            holder = (ViewHolder)row.getTag();

       final Book entry = (Book)super.getItem(position);

       holder.bookName.setText(entry.name);

       holder.chapterSelectButton.setOnClickListener(new View.OnClickListener() {

           public void onClick(View v) {
              context.selectChapter(entry);
           }
       });

       return row;
    }
}
```

Book.java

```
package com.khasibiblekonkordant;

import java.util.ArrayList;


import android.net.Uri;
import android.provider.BaseColumns;

public class Book implements BaseColumns {
    public static final Uri CONTENT_URI = Uri.parse(BibleProvider.CONTENT_URI +
"/books");

    public static final String DEFAULT_SORT_ORDER = "id";

    public static final String ID = "id";
    public static final String NAME = "Book";

    public Integer id = null;
    public String name = null;
    public ArrayList<Lynnong> lynnongs = null;


    public Book(Integer id, String name) {
       super();
       this.id = id;
       this.name = name;
    }

    @Override
    public String toString() {
        return name;
    }

    public static Uri getContentUri(final int bookId) {
       return Uri.parse(BibleProvider.CONTENT_URI + "/books/" + bookId);
    }

    public static String getWhereClause(final int bookId) {
       return "id = " + bookId;
    }
}
```

Bible.java

```
package com.khasibiblekonkordant;
import java.util.HashSet;
import java.util.List;

import android.app.AlertDialog;
import android.app.ListActivity;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;


public class Bible extends ListActivity implements OnItemClickListener {
    private static final String TAG = "Bible";

    List<Book> books = null;
    @Override
    public void onCreate(Bundle savedInstanceState) {
       Log.d(TAG, "onCreate");

       super.onCreate(savedInstanceState);
        setContentView(R.layout.bible);
        getListView().setFastScrollEnabled(true);
        getListView().setTextFilterEnabled(true);

        books = BibleLibrary.getBooks(getContentResolver());
        setListAdapter(new BookAdapter(this, books));
        getListView().setOnItemClickListener(this);
    }

    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
       Log.d(TAG, "onItemClick");
       final Book book = ((BookAdapter)getListAdapter()).getItem(position);
             gotoChapter(book, 1);

    }

    public void selectChapter(final Book book) {
       int count = BibleLibrary.getChapterCount(getContentResolver(), book);

       if (count == 1) {
           gotoChapter(book, 1);
       }
       else {
           final String[] chapterNames = new String[count];
           for (int i=0; i<count; i++) {
              chapterNames[i] = "Lynnong " + (i+1);
           }

           AlertDialog.Builder builder = new AlertDialog.Builder(this);
           builder.setTitle(book.name);
           builder.setSingleChoiceItems(chapterNames, -1, new DialogInterface.OnClickListener() {

              public void onClick(DialogInterface dialog, int which) {
                  gotoChapter(book, which+1);

                  dialog.cancel();
              }
           });
```

```
         builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
           public void onClick(DialogInterface dialog, int whichButton) {
              // Canceled.
              dialog.cancel();
           }
         });

         builder.show();
      }
    }

    private void gotoChapter(final Book book, final int chapter) {
      Intent intent = new Intent(Bible.this, ChapterActivity.class);
      intent.putExtra(ChapterActivity.TITLE, book.name);
      intent.putExtra(ChapterActivity.BOOK_ID, book.id);
      intent.putExtra(ChapterActivity.CHAPTER, chapter);
      startActivity(intent);
    }

    @Override
    public boolean onKeyUp(int keyCode, KeyEvent event) {
      if (keyCode == KeyEvent.KEYCODE_SEARCH) {

         Intent intent = new Intent(this, SearchActivity.class);
         startActivity(intent);

         return true;
      }
      return super.onKeyUp(keyCode, event);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
      MenuInflater inflater = getMenuInflater();
      inflater.inflate(R.menu.books_menu, menu);
      return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
      switch (item.getItemId()) {
      case R.id.load_bookmarks_menu_item:
         loadBookmark();
         break;
      case R.id.delete_bookmarks_menu_item:
         removeBookmarks();
         break;
      case R.id.search_menu_item:
         Intent intent = new Intent(this, SearchActivity.class);
         startActivity(intent);
         break;
      }
      return super.onOptionsItemSelected(item);
    }

    private void loadBookmark() {
      Bookmarks bookmarks = new Bookmarks(this);
      bookmarks.loadBookmarks();
      final List<Integer> bookmarkListing = bookmarks.bookmarks;

      if (bookmarkListing.size() == 0) {
           Toast.makeText(this, "Ym don Bookmark ba phi saved", Toast.LENGTH_LONG).show();
      }
      else {
         final String[] bookmarkStrings = new String[bookmarkListing.size()];
         for (int i=0; i<bookmarkListing.size(); i++) {
           Verse bookmarkedVerse = BibleLibrary.getVerse(getContentResolver(),
bookmarkListing.get(i));
           Book book = findBook(bookmarkedVerse.bookId);
```

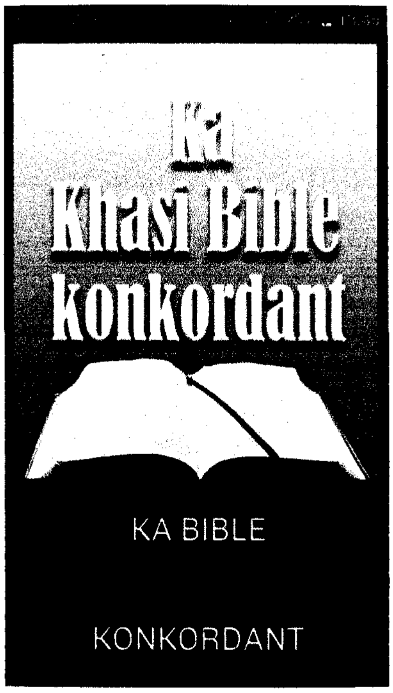
```
             bookmarkStrings[i] = book.name + " Lynnong " + bookmarkedVerse.chapter + " Verse " +
bookmarkedVerse.number;
           }

           AlertDialog.Builder builder = new AlertDialog.Builder(this);
           builder.setTitle("Load Bookmark");
           builder.setSingleChoiceItems(bookmarkStrings, -1, new DialogInterface.OnClickListener() {

             public void onClick(DialogInterface dialog, int which) {
                 Verse selectedBookmark = BibleLibrary.getVerse(getContentResolver(),
bookmarkListing.get(which));
                 Book book = findBook(selectedBookmark.bookId);

                 Intent intent = new Intent(Bible.this, ChapterActivity.class);
                 intent.putExtra(ChapterActivity.TITLE, book.name);
                 intent.putExtra(ChapterActivity.BOOK_ID, book.id);
                 intent.putExtra(ChapterActivity.CHAPTER, selectedBookmark.chapter);
                 intent.putExtra(ChapterActivity.VERSE, selectedBookmark.number);
                 startActivity(intent);

                 dialog.cancel();
             }
           });

           builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
             public void onClick(DialogInterface dialog, int whichButton) {

                 dialog.cancel();
             }
           });

           builder.show();
        }
     }

    private void removeBookmarks() {
       final Bookmarks bookmarks = new Bookmarks(this);
       bookmarks.loadBookmarks();
       final List<Integer> bookmarkListing = bookmarks.bookmarks;

       final String[] bookmarkStrings = new String[bookmarkListing.size()];
       if (bookmarkListing.size() == 0) {
           Toast.makeText(this, "Ym don Bookmark ba phi saved", Toast.LENGTH_LONG).show();
       }
       else {
           for (int i=0; i<bookmarkListing.size(); i++) {
             Verse bookmarkedVerse = BibleLibrary.getVerse(getContentResolver(),
bookmarkListing.get(i));
             Book book = findBook(bookmarkedVerse.bookId);
             bookmarkStrings[i] = book.name + " Lynnong " + bookmarkedVerse.chapter + " Verse " +
bookmarkedVerse.number;
           }

           AlertDialog.Builder builder = new AlertDialog.Builder(this);
           builder.setTitle("Weng noh iaka Bookmarks");
           final HashSet<Integer> bookmarksToDelete = new HashSet<Integer>();
           builder.setMultiChoiceItems(bookmarkStrings, null, new
DialogInterface.OnMultiChoiceClickListener() {

             public void onClick(DialogInterface dialog, int which, boolean isChecked) {
                 if (isChecked)
                    bookmarksToDelete.add(bookmarkListing.get(which));
                 else
                    bookmarksToDelete.remove(bookmarkListing.get(which));
             }
           });

           builder.setPositiveButton("Ok", new DialogInterface.OnClickListener() {

             public void onClick(DialogInterface dialog, int which) {
```

```
                                for (final Integer bookmark : bookmarksToDelete) {
                     bookmarks.removeBookmark(bookmark);
                  }
                  dialog.dismiss();
              }
           });

           builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
              public void onClick(DialogInterface dialog, int whichButton) {
                  dialog.cancel();
              }
           });

           builder.show();
       }
    }

    private Book findBook(final int bookId) {
       for (int i=0; i<books.size(); i++) {
           Book book = books.get(i);
           if (book.id.equals(bookId))
              return book;
       }
       return null;
    }
}
```

11. Sample Input/ Output Screens:

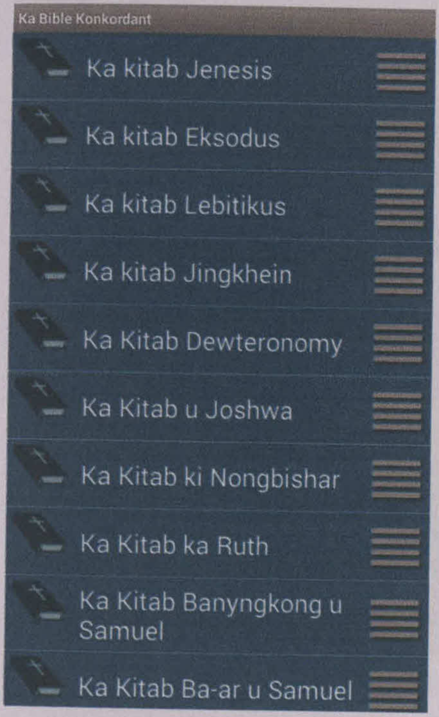
1: First page

In this activity you can choose whether you want to view the books of the bible or perform the konkordant that that is link from this konkordant button



Ka Khasi Bible konkordant

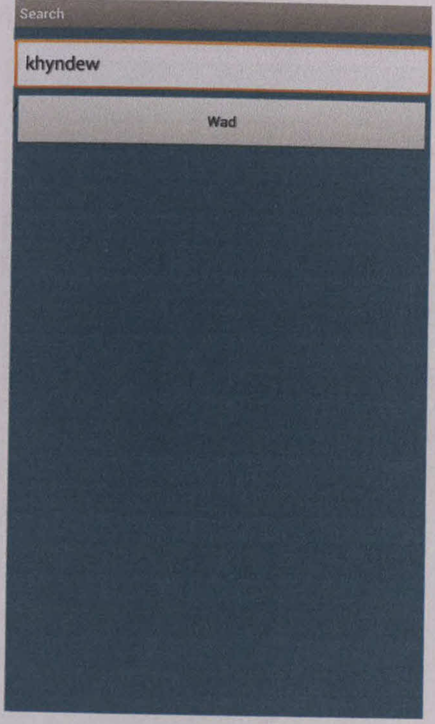
2 > Link from the Bible Button in this activity you can view the books of the bible.



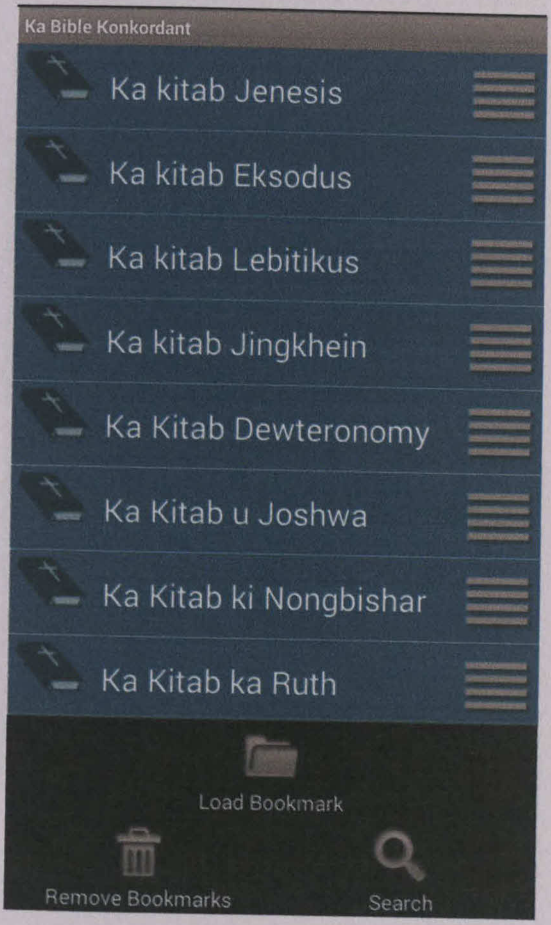
Ka Bible Konkordant

- Ka kitab Jenesis
- Ka kitab Eksodus
- Ka kitab Lebitikus
- Ka kitab Jingkhein
- Ka Kitab Dewteronomy
- Ka Kitab u Joshwa
- Ka Kitab ki Nongbishar
- Ka Kitab ka Ruth
- Ka Kitab Banyngkong u Samuel
- Ka Kitab Ba-ar u Samuel

3: Link from Konkordant in this acticity you can perform the search



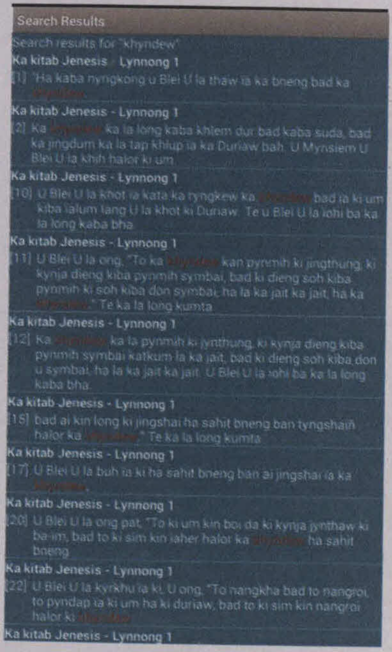
4 > In this activity you can perform the search and load the bookmarks saved



Ka Bible Konkordant

- Ka kitab Jenesis
- Ka kitab Eksodus
- Ka kitab Lebitikus
- Ka kitab Jingkhein
- Ka Kitab Dewteronomy
- Ka Kitab u Joshwa
- Ka Kitab ki Nongbishar
- Ka Kitab ka Ruth

5 > This is the activity that shows the results searched



Search results for "khyndew"

Ka kitab Jenesis - Lynnong 1

[1] "Ha kaba nyngkong u Blei U la thaw ia ka bneng bad ka khyndew.

Ka kitab Jenesis - Lynnong 1

[2] Ka khyndew ka la long kaba khlem dur bad kaba suda, bad ka jingdum ka la tap khlup ia ka Duriaw bah. U Mynsiem U Blei U la khih halor ki um.

Ka kitab Jenesis - Lynnong 1

[10] U Blei U la khot ia kata ka ryngkew ka, khyndew bad ia ki um kiba ialum lang U la khot ki Duriaw. Te u Blei U la iohi ba ka la long kaba bha.

Ka kitab Jenesis - Lynnong 1

[11] U Blei U la ong, "To ka khyndew kan pynmih ki jingthung, ki kynja dieng kiba pynmih symbai, bad ki dieng soh kiba pynmih ki soh kiba don symbai, ha la ka jait ka jait, ha ka khyndew." Te ka la long kumta.

Ka kitab Jenesis - Lynnong 1

[12] Ka khyndew ka la pynmih ki jynthung, ki kynja dieng kiba pynmih symbai katkum la ka jait, bad ki dieng soh kiba don u symbai, ha la ka jait ka jait. U Blei U la iohi ba ka la long kaba bha.

Ka kitab Jenesis - Lynnong 1

[15] bad ai kin long ki jingshai ha sahit bneng ban tyngshaiñ halor ka khyndew." Te ka la long kumta.

Ka kitab Jenesis - Lynnong 1

[17] U Blei U la buh ia ki ha sahit bneng ban ai jingshai ia ka khyndew.

Ka kitab Jenesis - Lynnong 1

[20] U Blei U la ong pat, "To ki um kin boi da ki kynja jynthaw ki ba-im, bad to ki sim kin iaher halor ka khyndew ha sahit bneng.

Ka kitab Jenesis - Lynnong 1

[22] U Blei U la kyrkhu ia ki, U ong, "To nangkha bad to nangroi, to pyndap ia ki um ha ki duriaw, bad to ki sim kin nangroi halor ki khyndew.

Ka kitab Jenesis - Lynnong 1