

Politehnica University of Bucharest

Computer Science and Engineering Department

DynamicLib

Eftenoiu Alina

Focşa Oana-Cătălina

Iosif Paula-Alexandra

SSA

Bucharest

2015

Contents

1.	Project summary	4
2.	Project objectives	4
3.	Activities diagrams	5
3.1	Server activities diagram.....	5
3.2	Client activities diagram	5
3.3	Location service activities diagram.....	6
4	Work plan	6
5	Risk management	10
5.1	Risk identification.....	10
5.2	Responses to risks	10

1. Project summary

The project consists of implementing a dynamic library which searches for the nearest library that offers the book requested by a certain user. The search explores the Humanitas, Carturesti and Diverta libraries.

The server is a PHP application which authenticates and registers a user with a specific password and username, searches for a book name and then stores a particular history for each user search and a global book rating into a database. The information provided is easily accessible by any client instance with GET methods through XML responses in a standard format, as presented in the further sections.

The client is an Android application written in Android 4.1.2, SDK revision 16. It has a simple and friendly interface and consists of two parts. The first part is the communication with the server and the second one is getting user location and calculating the nearest library.

The user can easily login using a username and a password or create a free account by introducing the credentials requested: username, password, email. The user also has the option to try out the application as a guest. If the user is logged in, he also has access to a list of previous searches. After he selects one of these options, the user will encounter a new page where, by simply introducing the name of the book he wants to search, will receive information about the book in every of the 3 libraries: availability, price and rating. Based on the pieces of information received, the user will decide which book suits him best and locate it.

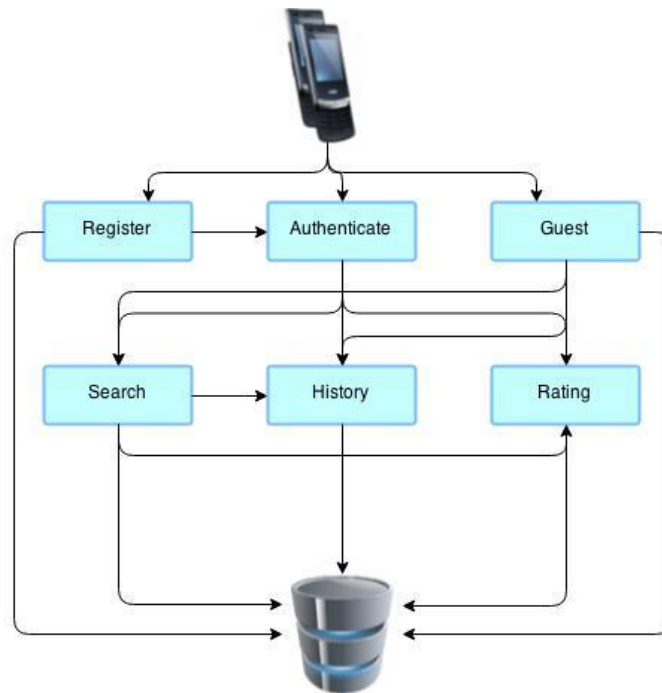
When the user wants to find the nearest library, a list of cities with the available libraries appears. When tapping on a city, a search will be made on Google Maps for all the three libraries. The search is transparent to the user and looks like this: „**Carturesti, Bucuresti**”. The user only receives the nearest library for each bookstore along with the address and distance.

2. Project objectives

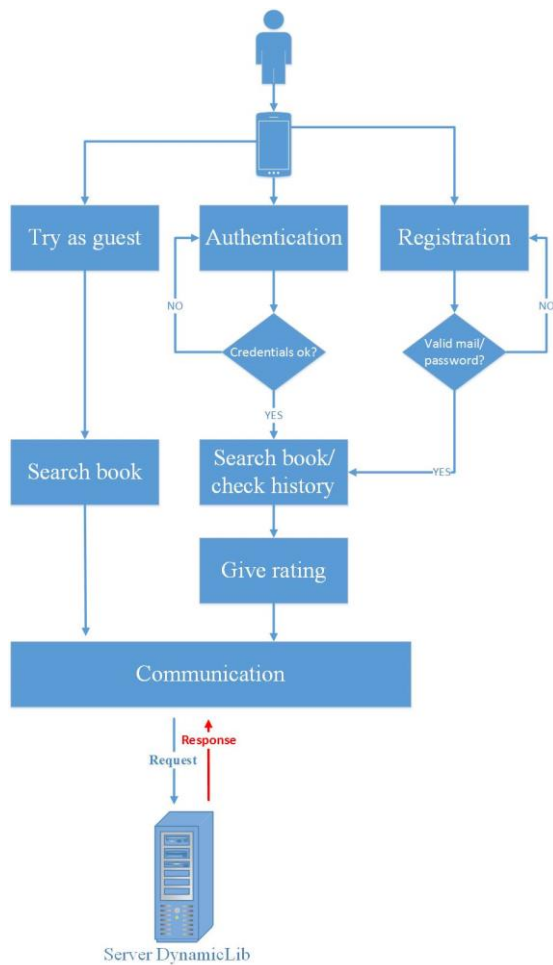
The objective of this project is to help users find any books that they are searching for fast and easy. This way, precious time and effort will be saved.

3. Activities diagrams

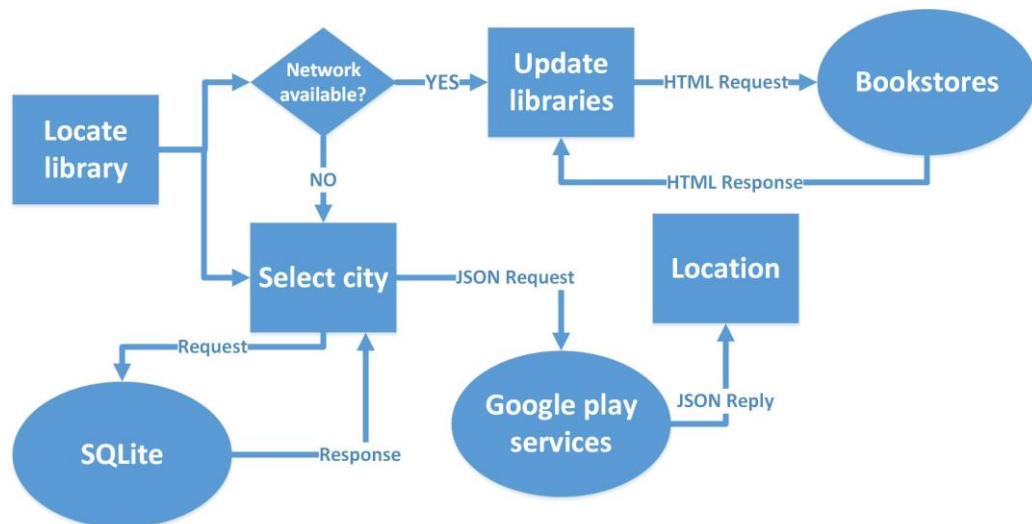
3.1 Server activities diagram



3.2 Client activities diagram



3.3 Location service activities diagram



4 Work plan

Server side – Oana-Cătălina Focşa

I started by looking for some parser examples in PHP and for various operations with a database.

I established the structure of the application and how the operations with the database are made and the conditions which determine them.

The architecture uses the following modules: authentication, register, history, rating and search which communicate with the database in various ways.

When trying to register, the server checks whether the client has provided valid username, password and e-mail.

When trying to authenticate, the server checks whether the client has provided a valid username and password, already registered in the database.

The user cannot login again if it is already logged in, it must logout first.

When searching for a book-name, the server must be provided a valid username and password for adding the search result to a history and the book-name into a common table from the database for further rating.

The server uses the search engines provided by the three libraries for each query. The results are the closest found to the queried string for each library, calculated with the levenshtein distance.

When trying to rate a book, the server must be provided a book-name and a value for rating, but also a valid username and password because only authenticated users are allowed to rate a book.

For guest access, the server has a user guest which is easy accessible from client side with username guest and password guest.

The architecture is simple, connecting the modules above to the database and taking queries according to the information needed.

The results obtained from queries are then provided in an XML format.

I tested the application in Google Chrome, by simply providing various queries and examining results in an XML format.

Client side – Paula-Alexandra Iosif

I started by looking for some of the native applications for Android existing or in development which are similar and I read some articles.

I established the structure of the application, how the pages are linked and the GUI of the application.

The architecture used by the two entities that communicate is REST (oriented resources), based on common language XML, the engine that makes it possible to transfer data via the Internet and HTTP common transport protocol. For implementing the functionality of the application, first of all it was necessary to establish the communication between client and server. For this, I used serialized objects Message type using HTTPConnection and URLConnection.

Urls and xmls:

- For authentication:
 - URL: <http://dynamiclib.host56.com/auth.php?name=user4&password=pass4>
 - XML:

```
<auth>
    ok
</auth>
....
```
 - Result can be “ok” (authentication was successful) or “notok” (username or password incorrect)
 - For “login as guest” option, username and password are always “guest”.
- For register
 - URL: <http://dynamiclib.host56.com/register.php?name=user2&password=pass2&mail=user2@yahoo.com>
 - XML: ...

```
<register>
    ok
</register>
...
```
 - Result can be “ok” (registration was successful), “toosmall” (password length < 4) or “notok” (incorrect email)

- For search book
 - URL: <http://dynamiclib.host56.com/search.php?name=user3&password=pass3&bookname=morometii>
 - XML: <bookname>


```

          <library1>
            <name>
              Morometii Vol.2
            </name>
            <availability>
              onstock
            </availability>
            <price>
              156,00 Lei
            </price>
          </library1>
          ....
          
```
- For previous searches
 - URL: <http://dynamiclib.host56.com/history.php?name=user5>
 - XML: <history>


```

          <name1>
            Nostalgia
          </name1>
          <name2>
            Morometii
          </name2>
          ...
          
```

 - It returns all previous books user5 searched before.
- For rating a book
 - URL: <http://dynamiclib.host56.com/rating.php?name=user6&password=pass6&bookname=morometii&rating=2>
 - XML: ...


```

          <rating>
            3
          </rating>
          ...
          
```

 - The result can be "notok" (values for rating must be between 0 and 5) or a numeric value (average of given value and the value before).

The basic functional unit of the system I used is Android Activity: the activity present in the flow of operations that make up the application. Each of the classes used in the application corresponds to an activity. Basically, as the application accesses resources, new pages are created. Elements on the pages (buttons, lists, TextView's) are described in the XML pages associated.

When testing the application, I initially used a virtual Android device (AVD) that successfully shapes a real device. Subsequently, the client application was installed on a mobile device smartphone HTC running Android OS version 4.1.1

Given the fact that the application is composed of multiple modules with different functionalities, I first conducted independent testing of components. Subsequently, I integrated all of these components to make up a functional system. To detect errors, I used the utility DDMS (Dalvik Debug Monitor).

Location analysis - Alina Eftenoiu

I started by looking for LocationServices API provided by Android and Google and I read a few articles and forum about how to get your current location on Android and how to search the map for other addresses.

I searched for Android interfaces, activities, ListView and all of the APIs needed to implement this. I draw a draft diagram of activities and decide how the application will look like.

I wrote the code and continued to document about the little stuffs that escaped on Analyze step.

I used Android 4.1.1, SDK 16 revision (4.1.2) and ADT 23.0. I wrote the code in Eclipse Luna with Android and Google SDKs.

I used google play services for getting the location and HTTPGet/HTTPPost to acquire a JSON with the specific coordinates and address of a given string to search on Google Maps.

From Android I used Button, ListView, TextView and other perspectives to create a user-friendly interface. The application check network connection and location service status.

I also used JSoup to parse html content from libraries websites. The addresses are saved in a database (SQLite Database) and are updated only when the user wants. Once I did this, the application works even if there is no network connection using the last know location and the data from the database.

The application allows the user to choose which city he wants to find the library. Then it displays he's location and the nearest locations for each bookstore in that city.

I tested the application on HTC ONE S running Android 4.1.1 and Nexus 7 running Android 4.4.1. The application had good results on both devices.

5 Risk management

5.1 Risk identification

If the components used, such as frameworks and other software modules, run with full privileges, an attack can facilitate serious data loss or server takeover.

If the server redirects users to other pages and websites, without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

If the client cannot connect to the server, the application won't be functional.

If no Internet connection is established, the location cannot be provided.

If the location service is off when running the application, it exists the risk that the program will consider the last location registered on device and this could lead to wrong output.

If the webhost we are currently using is not available, the server is not functional. We are not currently providing a local database for the application.

5.2 Responses to risks

Technology risks - Check how these performed in the past and we will know they will work.

Complex tasks - Decompose complex tasks into smaller tasks.

Using components with vulnerabilities - Inform about the components before using them.