FACULTY OF SCIENCE AND ENGINEERING

SEMESTER 2, 2017

IAB330: Mobile App Development

Assignment 2: App Prototype

Due Date: Monday, 30th October 2017, 5pm

Assignment is submitted as a team through Blackboard

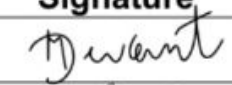
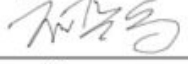
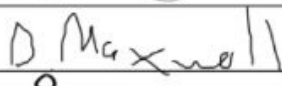
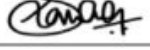
Weight: 40%

DECLARATION

You must sign below. By signing this form, you agree to the following:

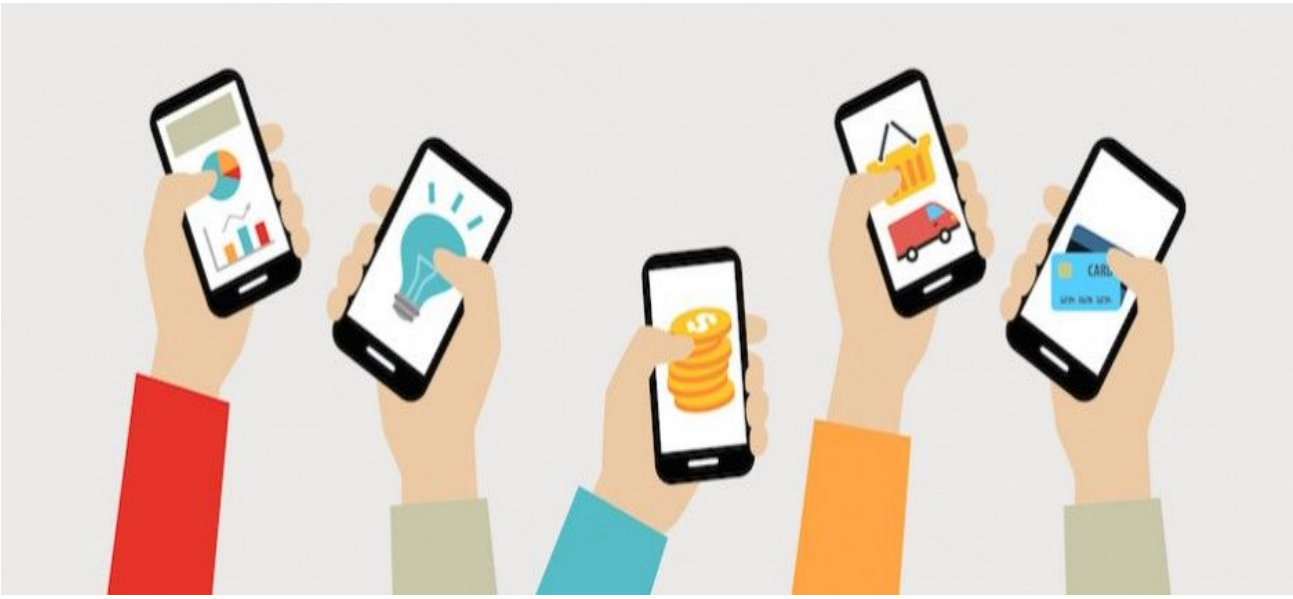
- We declare that all the work submitted for this assignment is our own original work except for material that is explicitly referenced and for which we have permission, or which is freely available (and also referenced)

The assignment shall be conducted in a team of 4 students, each team member must sign as it is a formal agreement that represents that everyone is contributing to the whole assignment.

Team Member Details		
Student Number	**Student Name**	**Signature**
9932305	Marie Devant	
5052165	Woong Jekal	
9670262	Daniel Maxwell	
9878831	Sandali Deenalattha	

FINAL REPORT

MOBILE APPLICATION DEVELOPMENT - IAB330



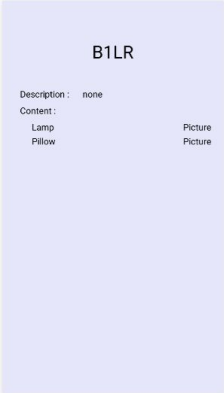
Marie Devant(9932305),
Woong Jekal(5052165),
Daniel Maxwell(9670262),
Sandali Deenalattha(9878831)

User Interface

Explanation of the User Interface

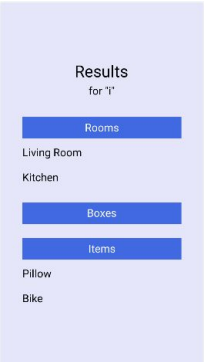
Include screenshots of the user interface of each page and explain the functionalities of your application

The main page of our application is the display of the current Move out. The boxes are displayed inside the room they belong to and clicking on the “plus” will lead the users to a new page, the detailed box.
The Search Bar and the Menu are available in the upper zone of the Title Page. Clicking on the button “Menu” will display the side menu and clicking on the search bar will display a keyboard to enter the string to search in the name of the objects.

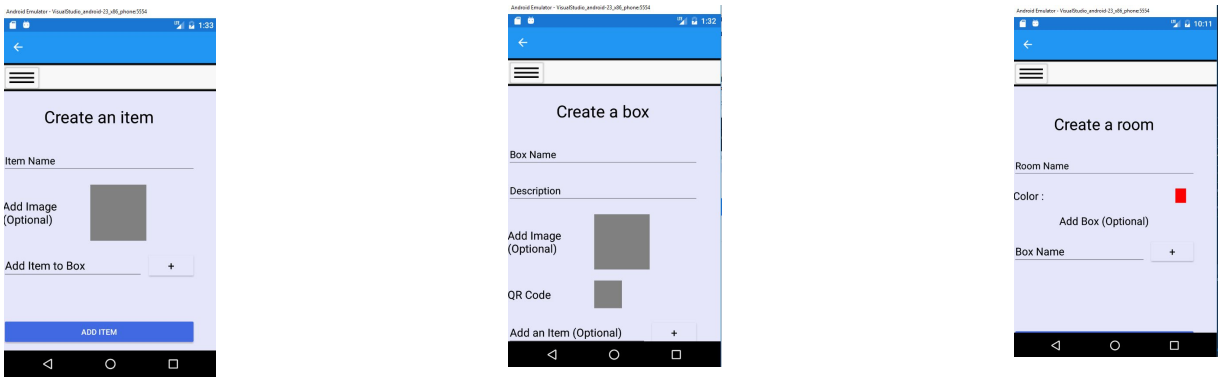


In the detailed page, all the items of the box are displayed, with a picture and a description for each box

Clicking on the Menu button, a side menu will be displayed, containing all the pages reachable from the current page.



The page Results is used to display the result of the search function. For a string inputted, the application displays all the objects (room, boxes, items), containing the string.



Pages of creation of any objects

The goal for this kind of pages is to allow the user to input the characteristics of the object he wants to create and then create and store it into the memory of the application.

Remove or Move Object Pages

The Move Page is used to move Items or Boxes from a Room/Box to another. The Remove Page is used to delete an Item or Box from the database.

Features of the application

Explain how your app implements each of the required and 'nice-to-have' features listed in the project specification.

- Ability to enter information about items and the box they are stored in:
This feature is ensured by the pages Create a Box/Items.

- Searchable database which shows a list of items in a particular box:
This feature is ensured by the page Detailed Box, when the user click on the "plus" button in the Main page, where all the rooms and boxes are displayed.

- Ability to list all boxes and all items in a room:
This feature is ensured by the Main Page, where all the boxes are listed and displayed. Display the items with a click on the + button.

-Categorize items by Room/Function:
In the class we created, there is a possibility to enter the Owner of an items. This assures the traceability of an item, we know what box it belongs to and therefore wht box.

-Moving: Group number by destination room (e.g. boxes 1-3 are for kitchen, 4-6 are for living room…):
We decided not to implement this feature as our boxes only have a name and not a number.

-Ability to move items between boxes or a box between rooms:
This feature is ensured by the page Move Page. This lets the user to move the item to another box if it's no longer in its original box or if a box is moved to a new room.

-Attach an image of the items in a box:
Even if the page Create a new Item owns a specific area to allow the user to enter a picture in the characteristics of an Item, the functionality could not have been implemented due to a lack of time.

-Ability to navigate through the application without any effort:
Side menu and page carousel feature helps to achieve this functionality.

-Ability to remove an item or a box:

This feature is ensured by the page Remove Page. This lets the user to remove an item or a box if it is no longer with the user.

Missing features

If you did not manage to complete the implementation, explain how you intended to implement the missing features.

QR Code

One feature of our project were to have option of having QR code. As the main functionality of the project became more difficult than originally estimated it was our options to delay this feature for later. Also, lack of time and understanding the implementation of the feature was also a problem.

Synchronize with several users

We planned at the beginning of the project to implement a way to share the database created in the JSON file. Like that, a simple mailing/share of the file would enable another user to use it. However, we had to change our implementation to SQLite as we could not figure out how to implement JSON correctly. From this we had did not have enough time to implement the synchronization feature for our mobile application.

UI patterns and considerations

Elaborate on the applied UI patterns, and explain your design considerations.

Navigation Between Pages

We decided to implement a side menu, that pops up when the user clicks on the menu button, available on all the pages. This is a really easy way to navigate between pages because if a user needs to skip pages without going through the whole process the user can use the side menu.

Page Carousel

The pages that are similar, like the creation an object pages or the Move/Remove Object are inserted into a page carousel. So for an example if a user is trying to create a box but forgets to creates a room, the user can go to Create a room page by swiping to left or right, without opening the menu again.

Realistic touch targets
We created buttons in pages, which is used when creating a move. The buttons were created in huge size so that the users can easily recognize the touch target.

Cut clutter
We did not fill up one page with many functionalities. We created separate pages for separate functions. Its neat and easy for a user to understand.

Simplicity
Our design is simple and easy to understand, anybody who is using the application can get a clear understanding of how to use the mobile app. This simplicity is even represented in the little number of color and always in the same tone.

Software Architecture

Description of the Software Architecture
Explain the software architecture of your implementation

The Model of data has been subjected to some changes from the first model created in the first report… Observing a little further the Room, Box and Item class, we decided to finally create a unique class allowing us to create an object, as it will be easier to handle. Thus, the class ToDoItem has the following attributes: Id, Name, Description, Picture, ObjectType (Room, Box, Item) and Owner that represents the Object owning the present Object.

Software Architectural Patterns

Elaborate on the applied software architectural patterns

The choice we made at the beginning of the implementation is to do an MVVM schema as architectural patterns.
Then, three folders have been created in the project. The Model folder contains all the class linked with the data Model (User, MoveOut, Room, Box, Item). The Views folder contains all the ContentPages (TitlePage, Create an Item/Box/Room, DetailBox, SearchResults, Remove/Move Object…).

Explanation of the components and Classes

Explain the implemented functionalities of each component and class

In the folder Model :
In this folder, a unique class implementing all the different type of object needed in our application is stored.

In the folder ViewModel :
The Sortage class has a function that allows the user to search the objects into the datas with a name containing a specific string. This methods is used by the search bar.
The class
The Mydatabase class contains all the required functions to connect with the sqlite database.

In the folder Image and Data:
These folders contain file useful for the design of the menu or the views and for the database.

In the folder Interface:
The folder contains an interface file. This interface allows the shared code to have access to specific functions situated in the code for a specific platform. As we only need to implement an application for an Android environment, the class that instantiate this interface is situated in the platform-specific project for Android. It is part of the function to open an access to the sqlite database, providing the path of the file, which depend of the environment of the Android application.

In the folder View :

The TitlePage is the main page, the one which is going to be display at the opening of the application. As the content of it is dynamic, depending on the database, the implementation is done directly in the .cs file in C#.
The Create Room/Box/Item ContentPage are the pages that are used to add an object into the database. The user is asked to enter all the details of the object, which will be used to create it in the database.
The MoveObject ContentPage is used to Move an Object (Box or Item) from a box or a room to another.
The RemoveObject ContentPage is used to completely delete an object (Item or Box) from the database.
The DetailBox ContentPage is used to display the items list of a specific box. You reach this page by clicking on the “plus” button in front of a box in the TitlePage.
The SearchResult ContentPage is used to display the result of the search function. After entering a string to search in the Search Bar on the TitlePage, the user will be redirected to this SearchResults page, where the results will be displayed in three categories : Rooms, Boxes or Items.

Testing and Quality Assurance Strategy

Quality of the application

Explain how you assured the quality of your application

Testing of the mobile application is not just to ensure the application quality but it is also is a measurement about how well the required features adhered. It also a continuous learning tools that will help process further improvements such as how do we make it lighter, how can we avoid similar bugs again. With the rapid changes in device software and designs, it is very important that application testing to not just assure the quality of the application but also the continuation of improvement in regards to ensuring the constant quality. The application needs to be tested every time where even a slight changes in a code has occurred and also new testing case should be developed in order to catch even more hidden bugs.

To help with our quality and communication during the development process we have used github repository as our code storage and we have opened group chat via Facebook to keep tracks of our work. In order to manage our non-error working build we asked the team for approval of pushing to the master after testing. In event of testing the new

functionality we used different branches to build our codes so that we could prevent any new error and duplication problem with the implantation of the codes. Once we found that it is working build without error we pushed the updated code to the master so further development can be made.

User experience is important

Testing Methodology

Our testing methodology involves the four major types of testing that are compatibility testing, integration testing, usability testing and unit testing. The compatibility and Integration testing are used to test the mobile application against many different devices and functions within these different devices. This testing was processed on Visual Studio using Xamarin Test Cloud which allows both automated testings but also allowing to check applications compatibility. The automated testing can be processed on many different devices in the variety of different form factors that are running popular Android and iOS Unit testing is used within development environment in which uses C# language to test and catch any errors but it also allows the developers for easier error handling while developing the mobile applications. Usability testing is all about the testing of different functions of the phone within the mobile application such as enabling GPS, powering the device on and off, camera function and other similar features that most mobile devices carry. Usability testing can also test more hardware process like checking CPU usages, Memory overload against resource usage of the mobile application.

Reflection on Learning

What we learnt

As a team, we learnt to exploit each quality and skill from the different members to share the tasks in an efficient and pleasant way. We also had to find the compromise between the very different way we individually use to work to respect a planning and deadlines. We have learnt as a team that even if each member were assigned separate tasks in different operation in the mobile application, it was necessary to have the based understanding of each other's tasks in order to keep the codes for quality assurance. We have realised that learning the new code language as a team is more effective rather than learning on our own. However, communication was good and everybody finally finds a place that suited them in the team. We also have learnt about the system architecture and it's architecture design patterns which we were able to implement to

our mobile application.

Challenges faced

Organizational Problems:

Emulators stopped working at the university lab and two of our member's computer was not able to run an android emulator due to their weak computation capacities. We had to distribute a new time for the tasks that required an emulator to those who were able to run it. Members without the android emulator had Visual Studio Previewer to create the views and the navigation function which was then handed to another member to test it. This, however, resulted in a huge amount of work for the only two members who were able to launch an emulator as we were not prepared for this from the beginning. This has resulted in a delay on the progress which delayed our implementation for the more complex feature for our mobile application.

Another thing we didn't think about when we made the first planning of implementation was the load of work from the other classes we had, obviously situated all at the same time of the semester… We were then a bit late for the first feedback and finally didn't take advantage of this first feedback as we should.

Implementation Problems:

During the implementation of the database, our first idea was to use a JSON file to store the data linked with a specific move out. We had the complex problem with the path we had to use to find the database in the project. We asked for help to the tutors but we didn't resolve our problem. Deadlocked, we finally decided to completely change our method and implement a database thanks to SQLite.

Secondly, we faced some issue about the various plugins that we have to use for our Xamarin project. As there was no clear guideline or documentation to which version or plugin we had to use to get that specific functionality working, it was very time consuming and frustrating process. For example, to have the SQLite functionality we need the plugin, but since there was too much plugin with the similar name and different author it was basically trial and error until we found the functioning plugin. This process was very frustrating since we needed to edit some codes every time we were trying different plugin as well as for as some plugin causing clashes with the android emulator.

Also, working with the Xamarin for the first time created some problem of

implementation for the functionality that our project wanted to achieve. Even though there was much documentation in regards to the various functions and implementation online, it was too broad and we could not gain much help from it. As there were too little details about specific functionality examples provided it was really difficult to implement the code to what our project actually wanted to do. In addition, there was no information that we could find useful for combining different elements within the Xamarin for implementation of certain functions so it made the development really difficult.

Prototype Working Video

https://drive.google.com/file/d/0B-50HR9I10OORWN2eTdiek9jYTA/view?usp=sharing