

KTH, School of Information Communication Technology
ID2216 Developing Mobile Applications

Project Report
Room of Requirement

Renzhi Xing(renzhi@kth.se)

Wei Wang (wew@kth.se)

Xi Zeng (xzeng@kth.se)

Yantian You (yantian@kth.se)

Mar 13, 2017

Table of Content

1. Introduction.....	2
2. Concept of Bringing to Market.....	2
3. Design Process.....	3
3.1. Brainstorming.....	3
3.2. Field Study, Interviews and Findings.....	3
4. Prototype.....	3
4.1. Sitemap.....	4
4.2. Clickstream.....	4
4.3. Balsamiq Prototype.....	5
4.4. Web Prototype.....	5
4.5. Android Prototype.....	6
5. Implementation, Methods and Technology Used.....	6
5.1. Front-end Development.....	6
5.2. Back-end Development.....	11
6. Potential for Business Development.....	13
7. Conclusion.....	13

1. Introduction

The final goal we have pursued during the project is to help people to find all they want in our platform, just as our group name “Room of Requirement” describing. Our main idea is to create an application that provides lost and found functionality for normal people. Targeting at people who lose something and would like to find it, as well as people who are looking for owners of things that they pick up, this application makes the “searching and matching” process easier and more effective, thus brings convenience to our customers. In order to gather more users’ perspectives and ideas, we started by an interview, in which we randomly asked our participants several questions.

After analysing the results, we discussed and decided the specific functionalities of our application. As discussed, the core functionality of this application is to offer the lost and found service, and in order to achieve that, necessary information need to be gathered. Users will be able to register as a member of the app by providing their personal information, which includes name, email address, phone number as well as password. All members will be able to log in/out of the application. Besides, the application will store information of every item. Having these information, the web application will perform following features.

For example, every time a user lose or find something, he/she can post for the item. Apart from the information of the item, the user also needs to indicate the condition (lost/found) of the item and the date it is lost/found. After that, the system will store the status and the post date of the item. If a user look for things that they have lost, she/he can simply post the information and wait for the response. She/he could also search for the item by category, and the system will display related posts, from which users may find their lost item. Besides, the status of the item will be tracked. It will be in “pending” state when it is post. When the lost and found parties get contacted with each other, the state will be change to “completed”. Also, after the item is delivered to the loser, it will finally become “found”.

2. Concept of Bring to Market

Considering the fact that lots of people have encountered the problem of losing something, our application will be favorable and supportive. It has highly feasibility of implement on people of all ages, which indicate that our app own bright market prospects. On one hand, we can propagandize our app by calling people’s attention to find lost article online or post their finding online. This can be done by advertising the app on some social platforms, such as facebook, Twitter, Instagram and so on. On other hand, we may take actions with below-the-line promotion. For example, we could stick up our advertisement on posters, billboards or subway.

But if the number of people using the app is minority, matching a lost or found item may get impossible. In case of this situation, we need to encourage users to post their finding items in early phases. This can be finished by paying individuals who post their found items, which is an investment of our app.

When talking about how we make profit, adding some advertisements for other products is a common way. In additional, when some items mean a lot to the owner who lost property, some would offer monetary reward to find it. If the lost and found item match successfully on our app, we could require

payment for our services. We believe that as long as more and more users finding their lost property on our app, it will attract more customers using our app.

3. Design Process

3.1. Brainstorming

Since the scope of this course project is broad, we spend some time thinking and discussing about ideas that we were going to implement for our group project. We defined some principles that we were going to follow about the project idea. Firstly, we would like to come up with a relatively new idea. It might be difficult for us to devise a completely fresh idea, however, it was possible to find an area that has not developed structurally and maturely. Secondly, our application should be able to attract users and bring them real services. There are various objectives of applications, like entertainment and working. We wished that through our application, people's life would be easier and more convenient. Besides, it would be great if our application could finally be put into practice and generate profits.

Following these principles, we finally decided to develop a lost and found platform that provides people with fast and convenient method of finding their lost items or looking for owners of items they pick up.

3.2. Field Study, Interviews and Findings

In order to have a more comprehensive picture of our application, we did a field study and interviews. The methodology we used to conduct the interview was questionnaire (appendix A) and audio recording. Our target participants were normal people that were randomly chosen, like workers, students and the elderly.

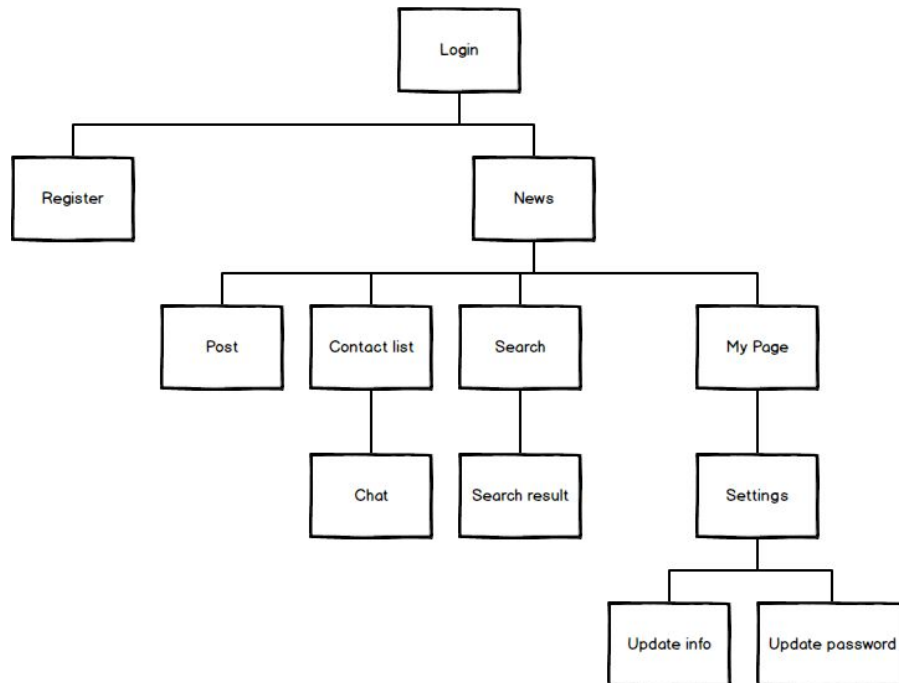
After gathering all the responses, we analyzed and found that almost all people have had experience of losing something and quite a number of them picked up something that were not theirs. Due to lack of methods of looking for them, only one third of them ended by finding the item. Also, another finding that is useful and meaningful for our idea is that for unnecessary items, people may not spend much time looking for it. However, when the item is of crucial importance, most people indicated that they would spend long time looking for it. Based on this data, we asked further about the approach they will take to look for or post the item. It was exciting for us to find that nearly half of our participants replied by posting the information of items on the social network and some of them were even willing to offer monetary reward.

Based on the findings we got from interviews, we further discussed and decided the features of our application, which will be discuss in detail in latter sections.

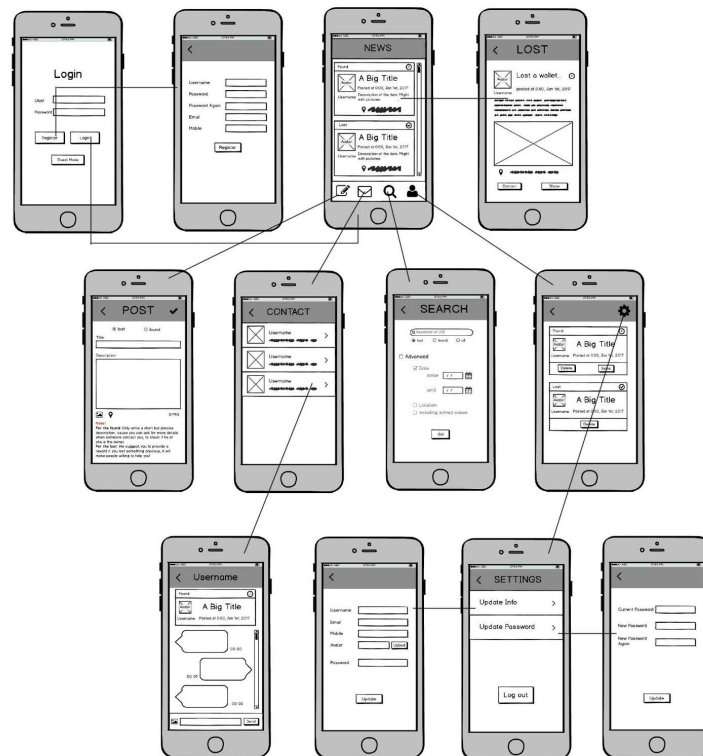
4. Prototypes

In this section, our sitemap, clickstream, balsamiq prototype, web prototype, android prototype and integrated prototype are displayed.

4.1. Sitemap



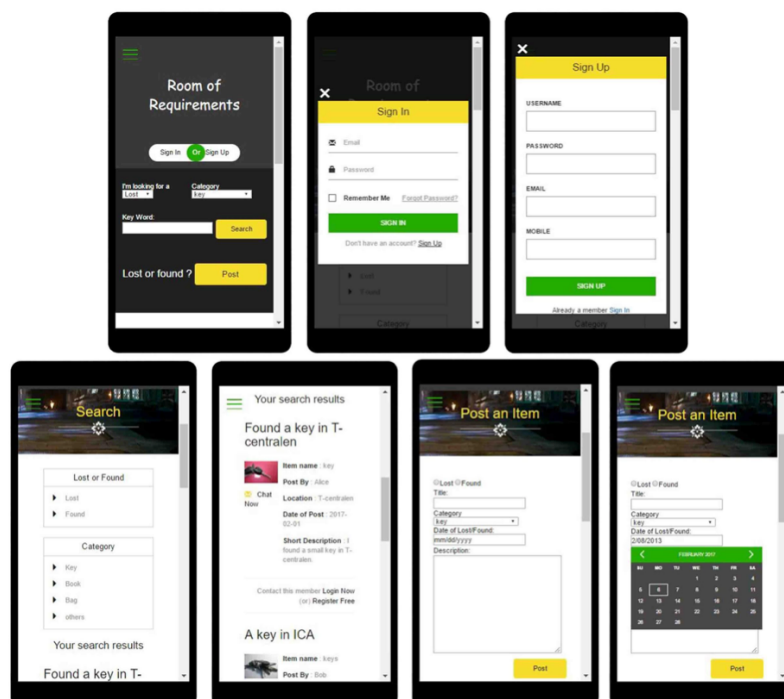
4.2. Clickstream



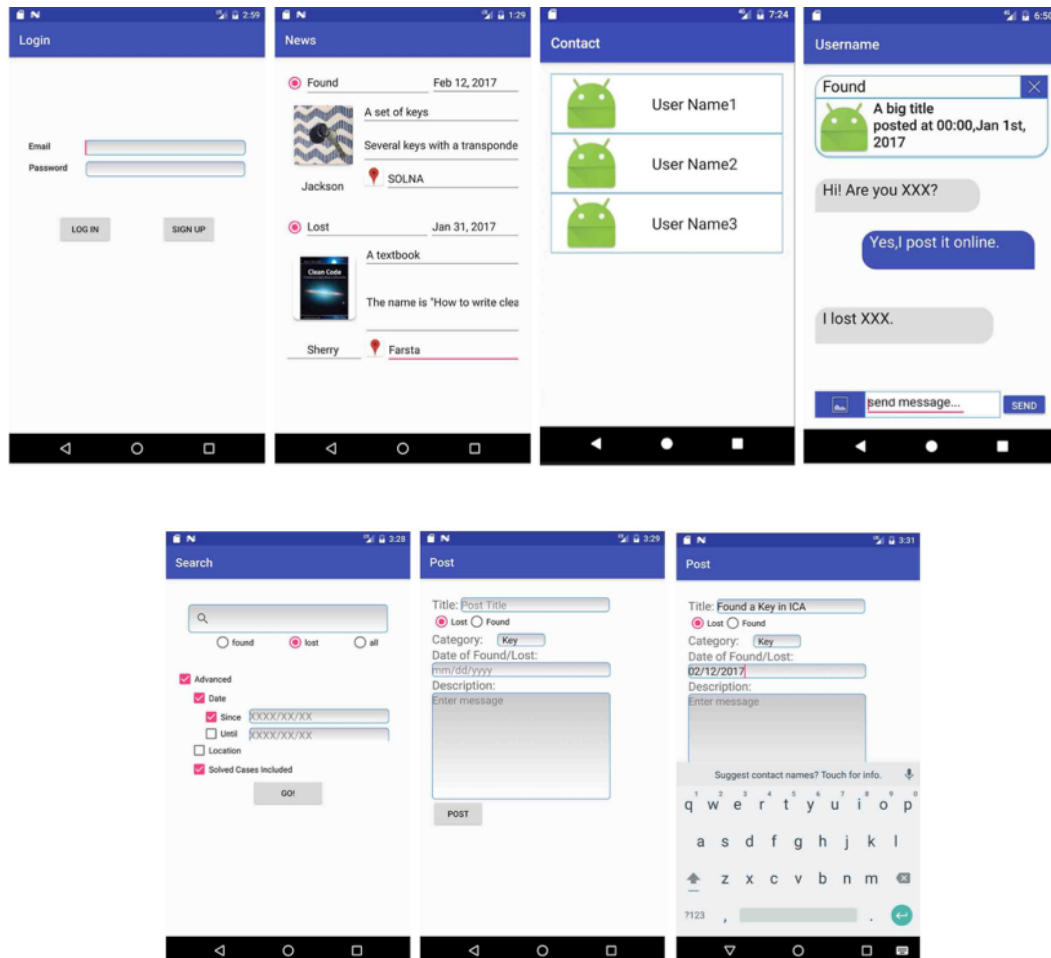
4.3. Balsamiq Prototype



4.4. Web Prototype



4.5. Android Prototype



5. Implementation, Methods and Technology Used

Since we developed our application in two groups, respectively working on frontend and backend, we will demonstrate our work in two parts.

5.1. Front-end Development

Tools to develop webpages are quite common and basic: HTML, CSS and Javascript. With no experience, we had to learn these languages from the beginning. Fortunately we found the work interesting and easy to learn. During the process of development, we got to know more techniques to achieve more complex functions, such as bootstrap and jQuery. The details of implements and the ideas behind them will be demonstrated as following.

Mainpage

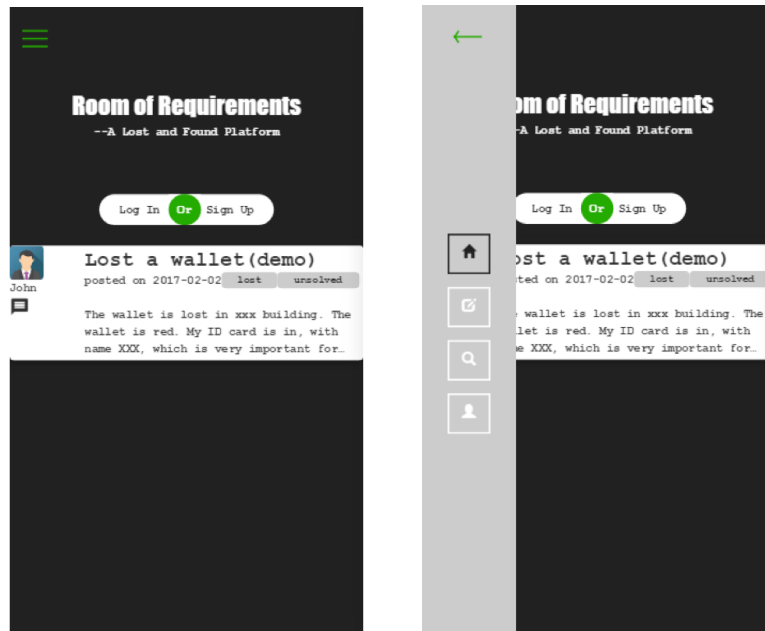


Fig 5.1 Screenshots of *Main.html*

Since our idea of this app is to build an online bulletin board, the main page of website is set to show all the posts in the order of date, from new to old.

A navigation bar is added to make it convenient to travel between pages. After clicking the icon at the top left corner, the bar will extend with an animation. Buttons on the bar respectively heading to main page, post page, search page and user's page (post and user's page are not accessible when the user hasn't logged in).

The animation of navigation bar is mostly achieved by CSS `@keyframes` rule, and `:hover` selector, which make the elements becomes bigger when being pointed.

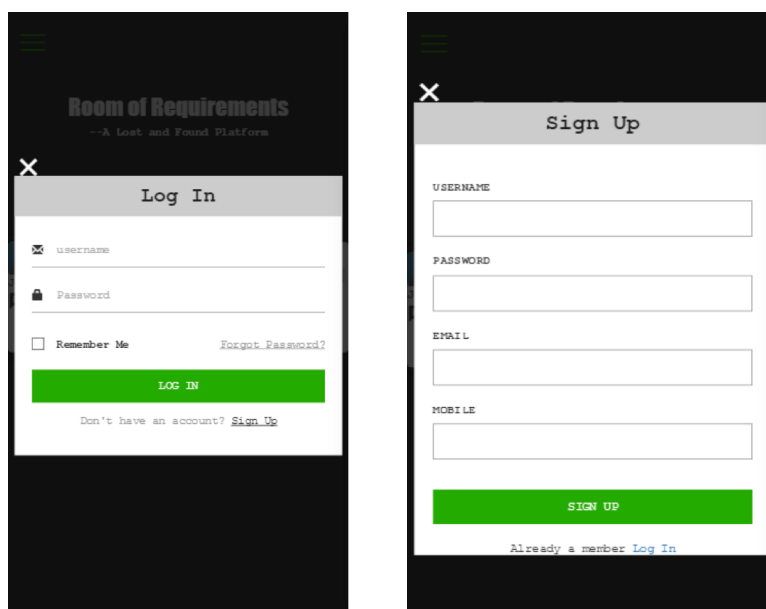


Fig 5.2 Screenshots of *Log in* and *Sign Up* small dialogs

When clicking at the button “Log In” and “Sign Up”, there will be small dialogs popping out, which is also achieved by CSS. When user signs up, they should provide their emails and mobile, which will let them receive notifications when there is a piece of news.

Post

The figure consists of two side-by-side screenshots of a web application interface. The left screenshot shows a form titled "Room of Requirements" with the subtitle "--A Lost and Found Platform". Below the title is a green button labeled "Post Your Item". The form includes radio buttons for "Lost" and "Found", a text input for "Item Title:", a date input for "Date of Lost/Found:" with a placeholder "yyyy-mm-dd", and a radio button for "I forgot.". Below these is a text area for "Description:". The right screenshot shows a "Photo Upload:" section with a button labeled "选择文件" (Select File) and a button labeled "Upload". Below this is a "Position:" section with a Google Map showing Stockholm, Sweden, and a "Submit" button.

Fig 5.3 Screenshots of *post.html*

This is the page where a user make a post of an item. The information we need includes:

- Whether item is lost or found.
- The title of the post, should includes brief and essential information.
- The date of lost or found. Considering the possibility of users forgetting the date they lose or pick up, a selection of “I forgot.” is given, and will set the date defaultly as the date of post.
- Description, the details addition to the title.
- Photo upload. Users can add one photo for each item. (As the opposition from group 3 referring, the words on the button is in the same language as the user’s system, which doesn’t matter at all.)
- Position. Here we introduced Google Map API to place a google map, and also the marker according to the current location of user, which can be moved to an accurate location of the item. This function asking for the location will work under the approvment of users.

After submitting, the informations will be stored in our database.

Search

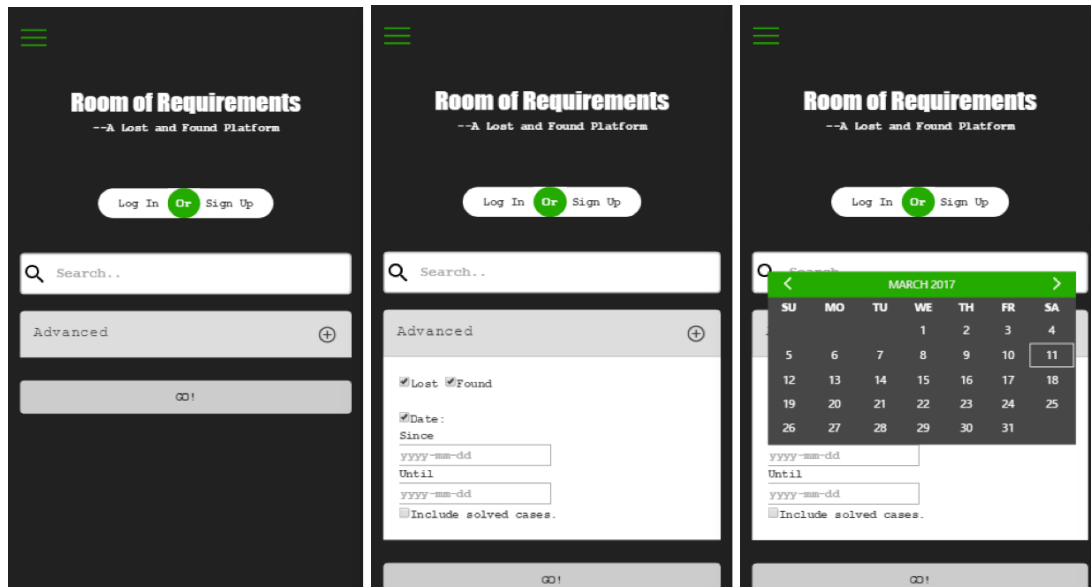


Fig 5.4 Screenshots of *search.html*

Another important function in our application is the search function. Users can search based on not only keywords, but also the attributes of items.

For a better user experience, we used an accordion area for the content of advanced searching. Also a date picker is inserted for user to choose a certain date instead of typing it, which can effectively avoid the user typing in datum with different format.

The functions are all based on Javascript.

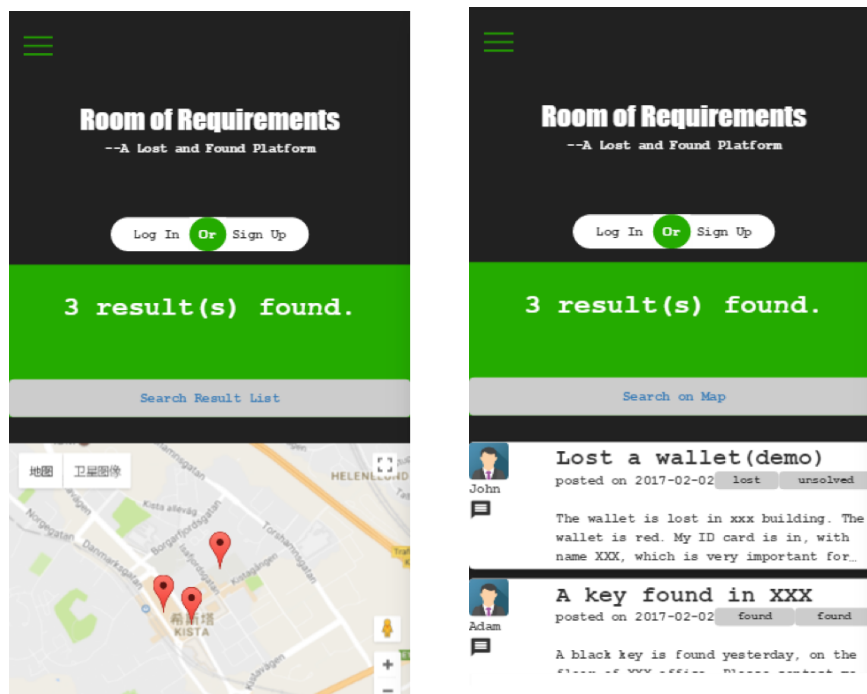


Fig 5.5 Screenshots of *search_on_map.html*

Search results will be sent back from back-end, with all the informations we want. We made two mode to show the results.

- List of items, which is similar to the main page.
- Results on map, which will show all the results as markers on Google map. The markers are able to be clicked and leading to the item details page.

Item Details

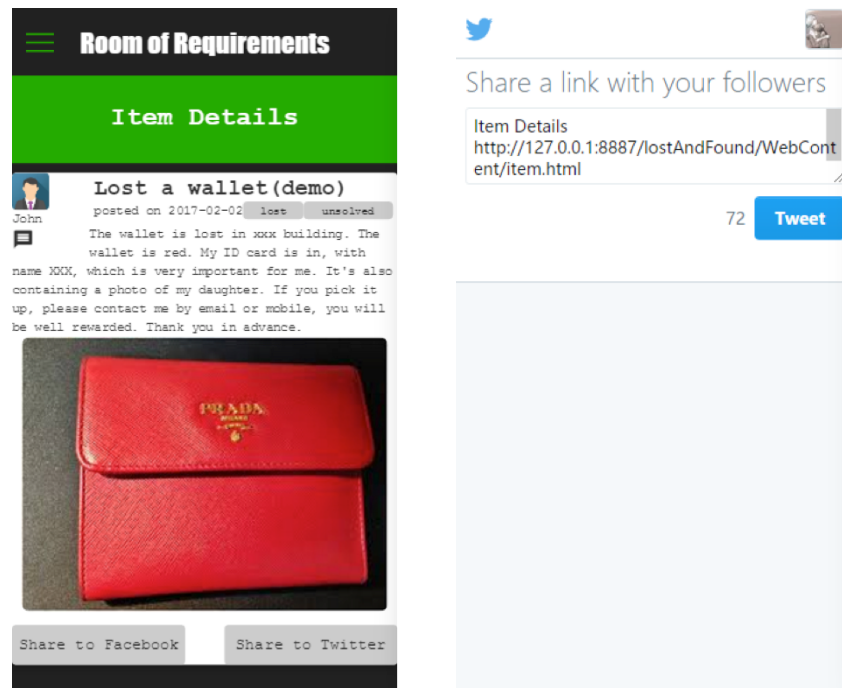


Fig 5.6 Screenshots of *item.html*

Item details can be reached by clicking the item cards in main page or list of searching results. It will show all the contents including the whole title, description and also the photo, if there is one.

At the bottom of webpage, there are two buttons used for sharing the url directly to other social networks, this will be useful if someone has something lost and is posted in our app, they will have more chance to find it even without knowing our app. This function is praised by our testers during the user test, and is also a brilliant way to attract more users.

Other pages

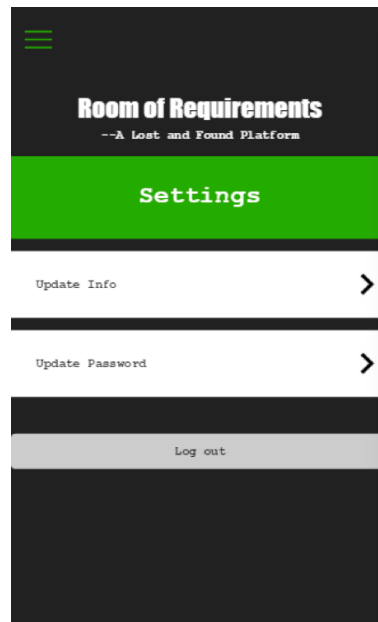


Fig 5.7 Screenshots of *settings.html*

Settings page is used for users to change their profile, password and log out. This is an essential part but easy to be ignored.

Interaction with backend

The interaction between frontend and backend was the biggest problem we met, since there are a huge amount of methods to learn about and choose from. In the end we decided to use Ajax based on jQuery functions to get and post data, with JSON being the data format sent and received during the interaction.

The reason why we chose Ajax and JSON instead of others that a professional website developer gave some suggestions on techniques, including Ajax and JSON, since they're broadly used in companies. It's better to use JSON because we worked mainly based on JavaScript and data in JSON format can easily be converted to JavaScript objects. As for Ajax, it is pretty easy to handle after we understood the theory. We also compared the differences between submitting form and using Ajax, and learnt that it works without refreshing the webpage, so we don't have to write extra codes to maintain previous page.

5.2. Back-end Development

We combined Java EE with Mysql to build our backend server based on Tomcat server and Struts2 framework. The software we used to build the server was Eclips and for database part, we used Mysql Workbench to manage our database. Each function in the backend server will extend the **ActionSupport** class and can be mapped into certain action in the **struts.xml** configuration file. **HttpServletRequest** has been used to achieve the data from frontend through **getParameter** function. For database part, we used JDBC to realize interaction with database. The database driver will be configured to establish the connection at the beginning. Then, **Statement** or **PreparedStatement** have been used to realize the basic SQL query or update. The results have

been stored into **ResultSet** and waited to be sent to the frontend after converted into Json format. The development of backend can be divided into several steps.

Establishment of database

We firstly create two tables, named **Item** and **User**, in our database. **Item** has been used to store information related to each lost or found item in our system and **User** has been used to store user related information. The following pictures show the structure of our tables in the database.

	itemID	property	title	description	photo	date	userID	stat
	1	found	A key found in XXX	A black key i...		2017-02-20	1001	unsolved
	2	lost	Did you see a purple ...	The wallet w...		2017-03-02	1003	unsolved
	3	found	test111	test111.....		2017-03-14	1001	unsolved
	4	found	Assignment 1 - Projec...	testing.....		2017-03-07	1001	unsolved

Figure 5.8 Columns in table **Item**

	userID	userAvatar	userName	password	email	phone
	1001	./images/img_avatar.png	Adam	123456789	adam@kth.se	0765833441
	1003	./images/img_avatar2.png	Shirley	123456798	shir@kth.se	076645321
	1004		Yantian	123456798	yantian@kth.se	13513856697

Figure 5.9 Columns in table **User**

itemID and **userID** are the primary key for table **User** and **Item**, respectively. **userID** is also the foreign key for table **User**.

Login/Logout and Signup

Session attribute has been considered to implement the login or logout fuction. During the login procedure, the user information has been kept by each session, the **SessionAware** interface has been implemented by **Login** function to store user information in the session scope while the **ServletActionContext** class has been used to achieve these information from the session scope. We also created a **LoginDao** function to realize the user verification during Login procedure. Once the user has logged into our system successfully, the current session will keep the basic information for this user for certain time until this user click the logout button or until session time has expired.

Compared with login and logout function, the signup function is much more simple. Once the user filled in the signup form and click the signup button, the frontend will send these data by using ajax to the backend server. Once the backend server received these data, it will check whether this user has been already registered. If not, server will store these user related data into our database for future log in and out procedure.

Post and Search

Post and search function can only be used when the user has successfully logged into our system. That is to say we need to check whether the current session contains valid user information at the beginning of these two function. If the verification failed, we will ask the user to login first.

The work after the session verification in these two function is quite similar to signup function. User filled in the required form in the frontend and data will be sent to the backend server. Backend server will then establish the connection with database and perform some basic SQL query to decide whether to put these data into our database. After that, backend server will send the selected data back to the front end to show them to users in the main page.

6. Potential for Business Development

As discussed in the previous sections, our group has discussed about the future development of our application to generate profits. Here are several approaches. Firstly, it is practical to implement a point-reward system in our application where the point could be transferred to real money and vice versa. There are several ways of earning points/money. For instance, when someone loses something important and the person is urgent to find it, he/she can reward some points/money. Part of the amount will be given to the person who finds it and the left part will be automatically given to our system. Secondly, we can enable advertisement in our application to earn money. Besides, another approach to earn money is to collaborate with other companies, particularly social media sites like Facebook and Twitter. This enables the synchronisation of contact information among them and thus improves efficiency and effectiveness, thus in the end attract more customers.

7. Conclusion

To sum up, with the initial idea of implementing a lost and found platform, our group intend to provide people with convenient and fast method to look for items they lost and look for owners of the items that they pick up. The sitemap, clickstream we implemented at the very first stage of development works as the foundation and based on them, we designed our web prototypes and android prototypes. Our group finally chose to develop a mobile web application that contains various features, including post, search, news, etc.

References / Scientific Sources

- [1] Fling, B. (2010). Mobile design and development. 1st ed. Sebastopol, CA: O'Reilly Media.
- [2] Lassoﬀ, M. (2015). Mobile app development with HTML5. 1st ed. Vernon, Conn.: LearntoProgram.tv.
- [3] Powell, T. and Powell, T. (2010). HTML & CSS. New York: McGraw-Hill.
- [4] jQuery.ajax() | jQuery API Documentation. [online] Api.jquery.com. Available at: <http://api.jquery.com/jquery.ajax/>.
- [5] W3Schools Online Web Tutorials. [online] Available at: <https://www.w3schools.com/> [Accessed 13 Mar. 2017].

Appendixes

Appendix A: Questionnaire for Interviews

1. Have you lost or found anything? If you have, do you remember when and where?
2. If you have lost something, was it important to you and did you find it finally?
3. How did you find it and how long did it take?
4. If you have found something, are they expensive and whether you find the owner of lost property?
5. How did you find him/her and how long did it take?
6. What will you do if you lost or found something?
7. Will you ask help by using related APP if you lost something?
8. Would you like to pay for finding your lost?
9. Will you find the owner of lost property using related APP if you find something?

Appendix B: Paper Prototype



Appendix C: Opposition Review Report from Group 5

General Feedback

From the past reports, we can see that your group have put much effort into this application. The reports contain many details of your project and help us understand better. Our group discussed and listed several good aspects about your work and also have some suggestions for further improvement.

Strengths

1. In your first assignment, which is the proposal, you elaborated the whole thinking process. It is really good to know your initial thoughts and how you did the research and interviews to finally decided the scope of your application
2. The user tests and feedbacks were well-written and explained in your report. You also made changes that. Some of the changes are quite effective. For example, the change you made to show the progress of the exercise and navigation button to the progress page.
3. In the whole process of application development, you keep paying attention on interview and user experience, during which you improve page design and usability.
4. You did a good job on dealing with detail. For example, you change the method of inputting Reps and Weight from a edit button to vertical spinners.
5. Name of your application is attractive and meaningful. “Exernice” indicates “Exercise” and “Nice”.

Suggestions for Further Improvements

1. In the final report, you might give more background information about the choice of developing this application(e.g. the amount of people who exercise). It will be good if you could address your focus of the application since there are many similar applications that are well-developed(e.g. fittime, keep).
2. We are a bit confused about the “promising to a friend” functionality. In which way should users promise their friends? What we can see from the web prototype is to click a “I promise” button and the notification will be sent to the person. However, how does this functionality attract users?
3. You mentioned in your proposal that some people like to use this kind of application because it is reward based. We suggest that in the future, if possible, you may add the point/reward system to attract more users. For instance, every time when a person achieves a goal or realize the promise that he/she makes to another person, he/she can earn some points.

4. We think it's better to have an adjustable timetable, exercise plan and the promise selections, better than choose a target from the list, since exercise works differently on people.
5. As an application, the functions are so simple that they can be achieved just by a reminder and sending emails. We have a suggestion that you can add notifications and widgets in the Android system, which is an advantage of native app.
6. Do you have any strength compared with other similar apps? What is your core competency?
7. Do you have any idea or plan about using this application to earn profits? And how?
8. When talking about "promising to a friend" functionality, you say "The user will first enter who to promise to by entering the person's email address or choosing one of their contacts.contacts. The contacts are based on the contacts of the phone and contacts from Gmail. " But you haven't point out how to realise it. How could your application interact with phone contacts? And will it sends a message to the friend?
9. You haven't point out what kind of message or reminder could be received, if a user friend acknowledge the "promising to a friend" email or message.
10. Till now, you haven't upload your assignment 4. So we couldn't see the technology and service you using on application development. Hope you give more technological information in your final report.