Roadies : Sprint Zero

Roadies is a roadside assistance application based mostly on mobile but supported by web services and a web application for the users. In the prototype, we have included the most important feature of the application “Repair” reporting i.e the user being able to login and report a problem. And, also for verification we have added the feature of being able to check their reporting information through the application website.

In order to implement the above mentioned feature in the application, we first decided on the technologies to be used, keeping in mind of its suitability in the final application. Android OS based mobile application and Java Spring Framework with MySQL database being the most significant decision.

Android OS is catching up rapidly in the phone market and provides good user experience through multitasking without keeping the application open at the same time. Also, it has more hardware options to choose from. Android technologies are open resource and can be developed in any platform unlike iOS application development. It is the only platform which provides easy to understand design environment. Each activity corresponds to a XML GUI and if any activity contains string value then it is stored in a separate XML file. Android development is based on Java and we have a team with Java skills, which also encouraged us in choosing Android as the platform for the mobile application.

In mobile, to implement the “Repair” feature of the application, we broke down the development into four signification section, which would give client a brief idea on how things run through and interact with server. Due to constraint of time we have implemented sections that are of utmost importance to represent the flow of the feature chosen. Also, the prototype was deployed on a local server so the accessibility is limit as of now and as per need further deployment will be done.

Login:
The login allows user to input a username and password and access the dashboard. Login function will accept any not empty input value and direct user to the main page. The information entered is used to identify a repair case.

Dashboard:
The dashboard activity is the main page of the mobile app that contains nine elements which allow user to choose a service. In the prototype, the “Repairs” button is active.

Repairs:
Repairs feature demonstrate how the problems get narrowed down to small criteria and allows the user to submit the detail..

Request:
Request is the sub section that deals with sends repair information to the server to obtain relevant solution. This sub section also involves GPS activity that gets the user’s current location. The solution is requested to the server through the web service provided by Spring Framework based web service. The part of getting response for the solution has not been done, because of its complexity that crosses the scope of the prototype.

For the web application, we decided on Java Spring Framework due to various reasons. The development base for the framework is large and would be easy for our developers to access support and information on the features of the framework. It is scalable and incorporates features to support REST based web services that will be highly useful in our mobile application to retrieve and submit data to the server. The development of the framework has a large group of experienced and skilled developers so the framework is reliable from errors and memory issues.

We were bound by time and resource constraints therefore we have given importance to display of user repair being reported to the server. The web application has the following features to support the user to verify if the server received their requests.

Login:
The user can login using the user information they provided in the application.

Dashboard:
The user can view the history of the cases he/she reported

The database we have used for the web application is MySQL, due to its strong stable history and being used widely in the web today. Also, it cuts down the expenses because it is free unlike paid services like Oracle and SQL server.

The above information clarifies the high level technology decision that we have made and the features we have provided in the prototype. In addition to that, we have chosen MVC architecture for its scalability, reliability, ease of development, success stories and providing more space for Separation of Concerns for future development of the application. UI design has been done in Photoshop. We hope you will have fun using the prototype!