Bilkent University

Department of Computer Engineering

Senior Design Project

Touravel: The Ultimate App for Traveling & Activities

High-Level Design Report

Gokhan Cetin, Gokhan Guler, Utku Bozoklu, Ekrem Dogan

Supervisor: Can Alkan

Jury Members: Fazli Can and Selim Aksoy

Progress Report
December 30, 2014

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

I. Table of Figures

1. Introduction

There is a highly populated community in the world: The traveling people. Traveling is a passion for some. Many backpackers have placed the idea of being on the road to the center of their lives and that is the motivation that directs them in their survival. With a slightly milder approach, considering not only the radicals but also all the outgoing and active people who like traveling; then there is an even greater amount of people to talk about. Sightseers, vacationers, travel bloggers, social media users, event-goers and much more - all those people are currently in a need of a technological solution to carry on their interest in the best way, which would solve their various problems using the state-of-the-art advances at the cutting edge.

People with high mobility have some requirements, constraints and preferences. There are many issues to think about such as transportation, accommodation, catering, and entertainment and so on when traveling is the matter. Even if there is some service to solve all the problems, there are some constraints with the device, like the mobility, power concept, accessibility, etc. Furthermore, as experienced travelers, we can state that some usability goals and other details should also be met in order to satisfy the demands of the active people. Touravel is the one to solve travel-related problems of such people.

1.1 Purpose of the System

Our proposed system aims to offer its users everything they would require regarding their travels during they are on the road. The purpose is to system is to provide some services that can be summarized in a few headings such as self-tracking, keeping statistics, sharing with social circles and getting advices from others about the trips that will be taken.

First of all, the system will enable the users to track every step they take and record all their achievements in a trip. The entire route as well as the photos taken, places visited and achievements unlocked will be logged and saved. Later, the data will be processed and will be served in a proper way after getting organized. Steps so far discussed are designed to be for personal usage and will serve as a traveling diary of the user. Starting from the next step, interaction will be allowed and encouraged.

In the following phase, the users will get the chance of being active in their social network with their traveling achievements. Any personal data collected and processed will be available for sharing. The sharing will be handled in different ways. Simply the data will be able to be shared in common social platforms, partially or fully. However, the main aim is to build the own social platform, designated with the specific purpose of being travelers network.

1.2 Design Goals

Usability: This can be thought as the main goal of the system. The system is designed to be simple and usable in every aspect. Considering that the user will be busy most of the time and will have limited time to interact with the portable device that the application will be installed on, the steps are mostly automatized. Even if the user would never contact the app, it will be able to keep working in the background. By this, the users will be enabled to focus on the travels and interact with the application only when they need.

Other than that, the menu will be designed in an easy-to-use way, and user-friendliness will be taken into account all the time. White space usage, for instance, will always be tried to exploited rather than having lines and other ways that could complicate the design and tire the users.

In order to provide the simplest design for the users, human-computer interaction concept will be considered all the time. Previous findings on the issue and conducted field study will be made use of.

Efficiency: This application may originally require a lot of energy. The reason for that is it will be working all the time and doing calculations that would consume a lot of power. Most specifically, usage of GPS may drain all the battery. Since the power issue is very crucial for current smartphones, special attention is needed for the efficiency issue.

To avoid power-related problems, we are going to use efficient algorithms and we will have different levels of usage to save energy when it is possible. For example, accurate data is needed when traveling actively and hanging out, but not, for example, staying at a place without doing much. Therefore, we will try to limit the power consuming by case-by-case treatments.

Additionally, the efficiency is needed to provide a fast application. The response time should be feasible and the performance should please the users. Since it would not be used if the application stalls and takes too much time to carry out certain tasks, we will ensure that it is not the case.

Effectiveness: We are claiming that the application will provide the users anything they would require during their travels. Then it should really do that. The application will carry out a number of tasks in a fast and stable way. Its users should be confident about being able to fulfill all of their requirements with the application, without any problems.

Portability: This is highly crucial and it is the reason for offering the application mainly for the portable devices. The target users will be in outdoor environments and naturally they need something to bring with them all the time. Consequently, the application will serve in a portable way.

1.3 Definitions, Acronyms, and Abbreviations

Server: A webserver that provides software services to clients.

Client: Software end that accesses a remote server on server.

HTTP: Hypertext Transfer Protocol.

REST: Representational state transfer.

1.4 Overview

Touravel comes with the claim of solving travel-related problems of vibrant people. Its purpose is to propose a system that provides many functions for its users such as self-tracking, keeping statistics, sharing with social circles and getting advices from others about the trips. While doing this, some design goals are considered to provide an application that would be favored by the users. Regarding this, the application should be usable, efficient, effective and portable.

2. Current Software

2.1 Chronos

Chronos [1] is a mobile application that tracks the user's daily activities. It basically saves all the places the user was at throughout a day. User can ignore, confirm or highlight a saved place and categorize it as home, work, social, exercise, education, etc. This helps to organize saved places. The user also can assign goals, such as 'go to gym for 30 minutes', 'work more than 8 hours', 'spend time with 2+ friends', etc. Goals are calculated automatically by Chronos according to activities. There is also option for adding friends to record automatically when and where the user is together with friends. There is an option to see places visited and routes between them on a map day by day.

Compared to our system, Chronos is generally focused on local use. People with daily routine lives are more likely to use this application efficiently. Types of goals and close attention to activities with friends in the interface show this. Our system is more focused on traveling records and recommendations. There is also no option for sharing on other networks in Chronos, which is available in our system.

2.2 Moves

Moves [2] is a mobile application that records how much distance is walked and which places are visited during this period. It shows a brief of the day as a timeline and the amount of walking in terms of distance, time and steps. The application can use the opportunities of the mobile phone to accurate information it collects. The user can share its activities on other networks. However, there is no 'friends' to follow or to tag and there is no a recommendation system.

Although it seems very similar to our system, Moves focuses on a completely different aspect of outing. For it, it is more important how much the user 'walked'. Places that are visited are rather used for milestones of walking for a day. In our system visited places are more focused and important to be tracked. In general; although they seem very similar in use, Moves is mostly designed as a health application while our system is designed for touring and sharing.

2.3 Rove

Rove [3] is a mobile application that records what the user has done at a time in a place. For any time of a day, users can add a note for the place for that time and save. The interface shows notes as a timeline. The sun's position is implemented as a circle at the top of the interface and tapping any point of this circle allows you to add a note to the corresponding time of the day. Below the sun circle, all past days are listed. Tapping one of them lists the notes and activities of that day. Users can also see the activities of a day on a map and share them on other networks.

Interface and usage of Rove show that it is designed for a personal note taker. It is similar to diaries. Features and interface don't provide using it interacted with other people. Users can track and save their activities but they are mostly saved only for the user. Socialization is one of the main points of our system and it is promoted as much as possible. Unlike Rove, in our system users can share activities, express opinion on them and ask recommendation for them.

3. Proposed Software Architecture

3.1 Overview

We have introduced the proposed software architecture of our project in the following subsections. Firstly, we describe our subsystem decomposition with packages, relations and dependencies. Then, we represent Touravel's software-hardware mapping. Data management and access policy are described in sections 3.4 and 3.5. Finally, we introduce global software control and boundary conditions in the last two subsections.

3.2 Subsystem Decomposition

Our system has general characteristics of client-server systems. It basically consists of a server that is in charge of data and loads of clients that have access to data through server. In Figure 1, you can see general view of the system.

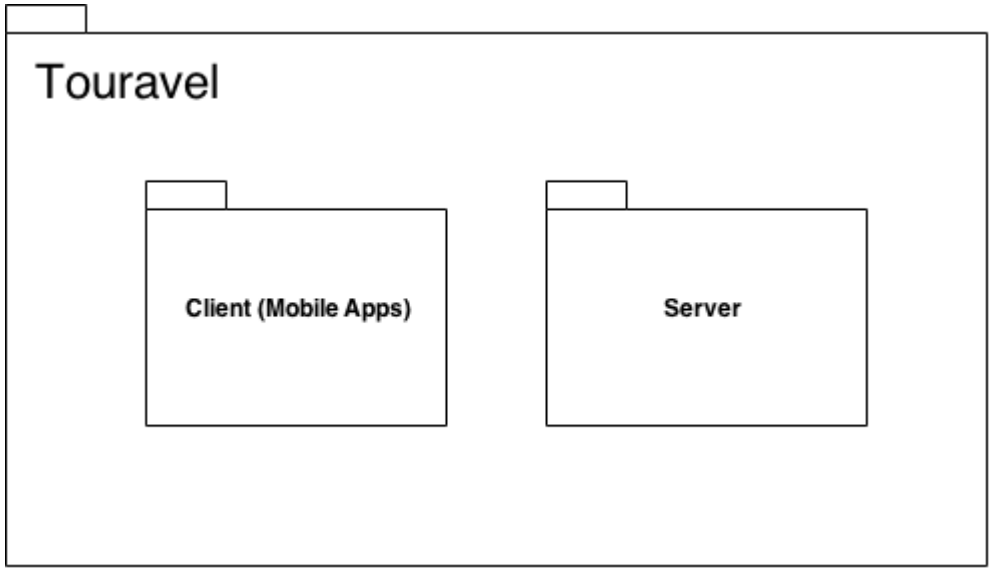

Figure 1 - General View of the System

As it is clearly seen in Figure 1, our system consists of Client and Server subsystems. Figure 2 shows decomposition of Client subsystem.

Information Tracker responsible gathering data from user's device. His/her whereabouts, route preferences, and social relationships. Information Tracker turns raw data into meaningful user actions for Business Logic to use.

UI Manager handles user interface components. There will be couple of screens that is available to user.

Communication Manager is responsible for communications with server. It will send requests and translates responses for Business Logic to use.

Business Logic is the brain of the client. It will run in the backend of the client. It will be responsible for organizing Information Tracker, UI Manager, and Communication Manager.

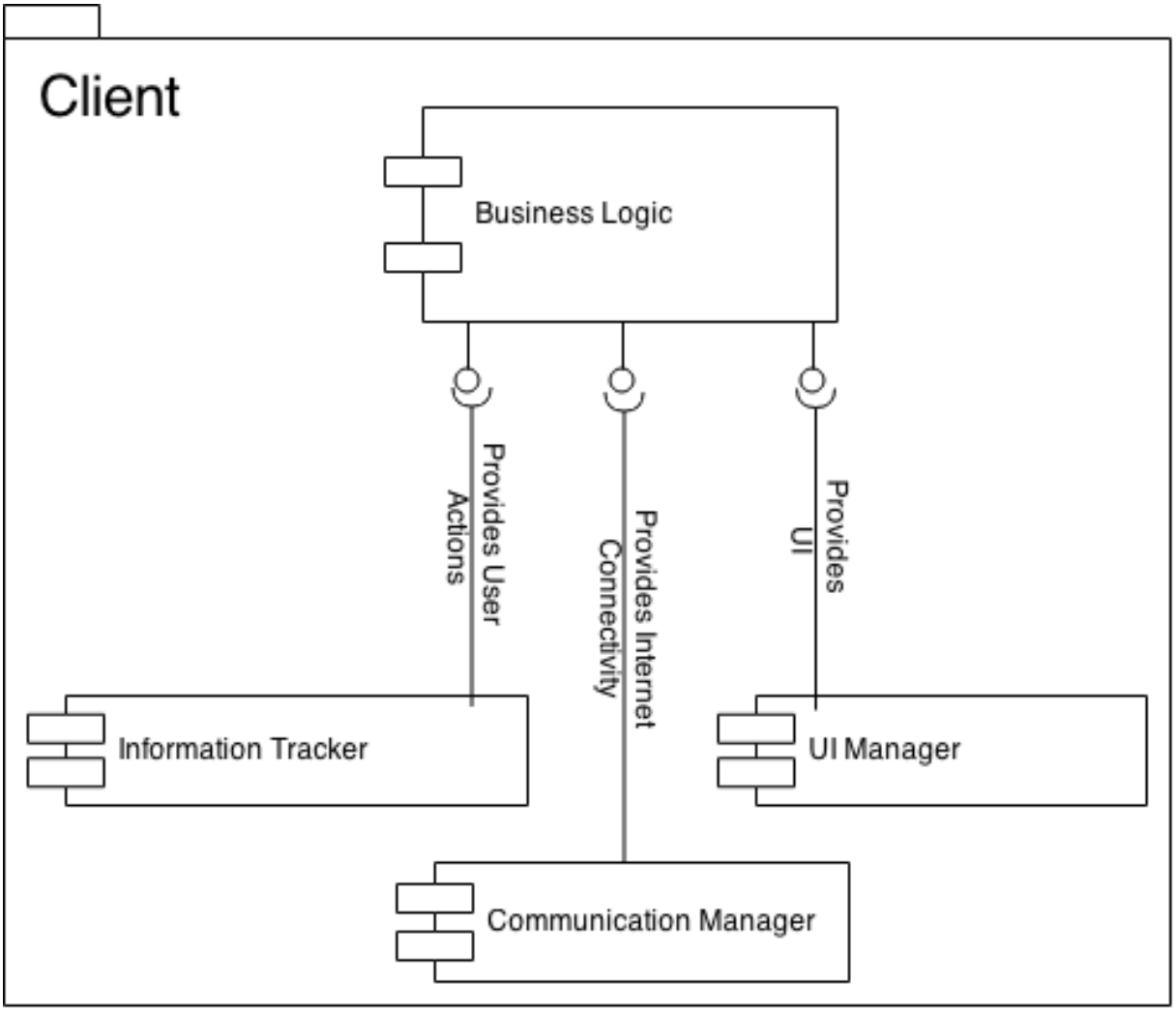

Figure 2 - Client Subsystem

Other subsystem of the Touravel is Server. Server is more sophisticated than Client. It has connection on the both ends; one to client and one to database servers. Server is responsible for correlate each data collected from different user; making meaningful assumptions out of them. Server, also will be responsible for managing data persistency.

As it is seen in Figure 3, server consists of Web services, Route Generator, Social Relationship Manager, Key-Value Cache Manager, Database Manager, and Key- Value Store.

Web services will be available to clients to communicate. HTTP based RESTful web services will be used. Authentication, data retrieval, data transfer are the main things web services will be responsible of.

Route Generator is responsible of generating new routes as requests from users. It has access to other people's routes from database.

Social Relationship Manager is responsible of social relations between users such as mentions and private messages.

Key-Value Cache Manager is responsible of handling key-value pairs for user. It will work under web services. It will keep track of mostly accessed key value pairs and store them in Key-Value Store.

Key Value Store will be a simple software that lives on primary memory. We are considering to use Redis for this matter; however, we may use memcached instead of Redis.

Database Manager is responsible of communication between Database and System. In cases of crashes, Database Manager will be the part which is responsible of database recovery.

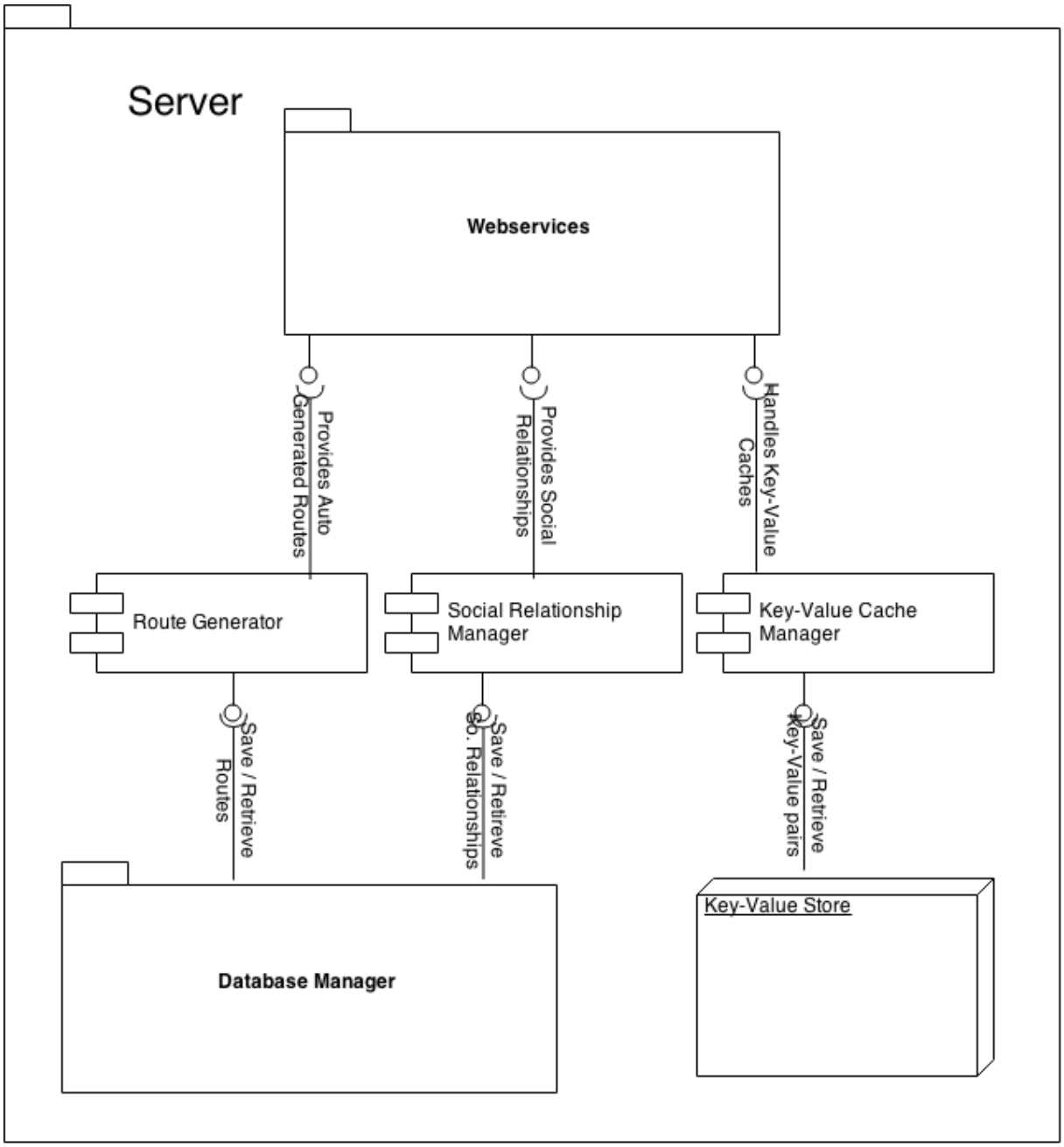

Figure 3 - Server Subsystem

3.3 Hardware – Software Mapping

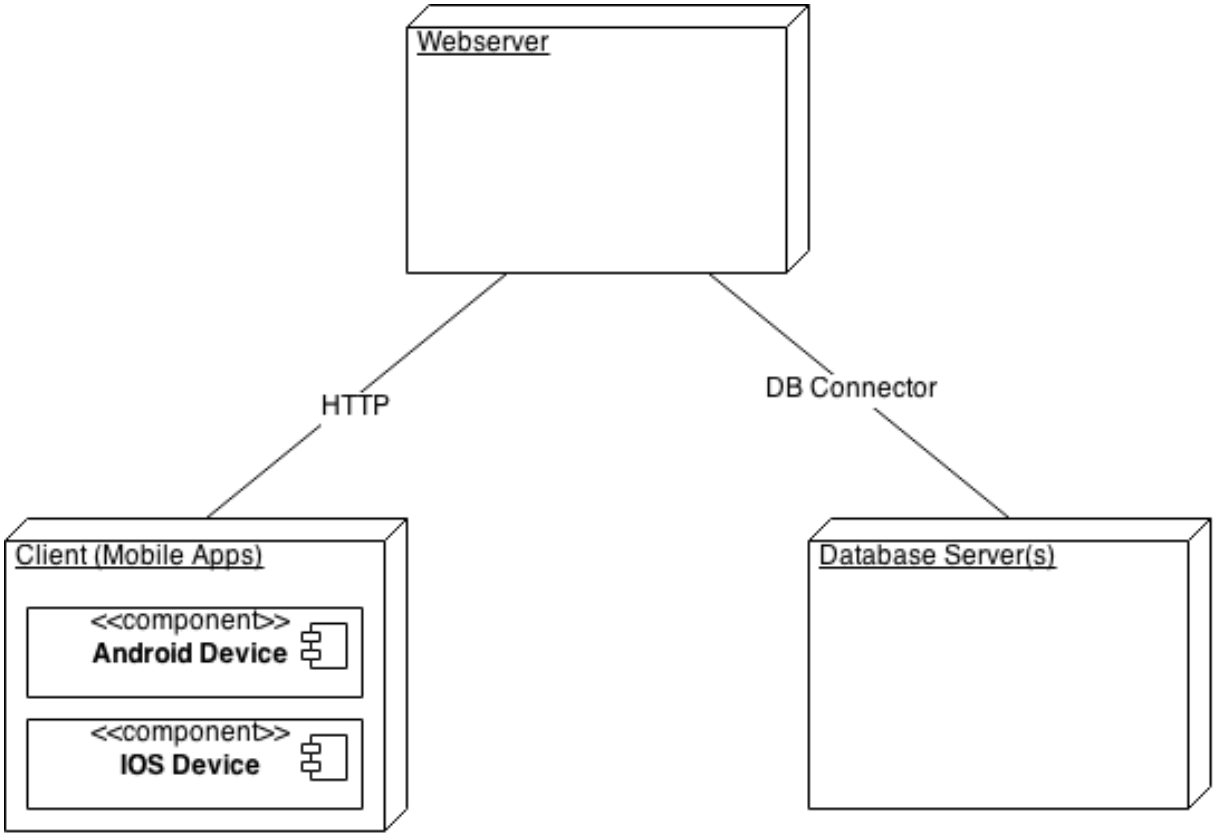

Figure 4 - Hardware - Software Mapping

Users will be clients for our system. They can either use Android or IOS Devices to access to the server. They will need internet connectivity to interact. Communication between Client and Server will be in HTTP protocol. Each user needs to download from mobile applications from application stores for free to be part of Touravel ecosystem.

Database servers will be in cloud. They cannot be accessed other parties than our webservers. We are going to use MongoDB since it has support for dynamic queries. It has advantages when partition between database servers occurs. We do not want our uptime is decreased because database partitions.

3.4 Persistent Data Management

Data Management is crucial for our system. There will be loads of database access; write/read requests. What we need is reliable, partition-tolerant database system. We choose MongoDB as it is indicated above. It is a Document Oriented Storage, it has full index support, and it has high availability. It also supports map/reduce; flexible aggregation and data processing.

User requests will be main reason for database access. Other than user requests; we are going to process user routes to generates new/better ones.

Details of database design will be discussed in Low-Level Design report.

3.5 Access Control and Security

Touravel has a membership system and highly encourages its users to register and use the system with their specific profiles to benefit from the features exclusive to members. Unregistered users would have access to very basic individual features only and will have no access to private data.

Registered users introduce themselves to the system by using login facility. All users need to log in before taking any action under their registered profiles. However, they might log in once and ask the system to remember their credentials from the same device for further usage. Every user's access will be limited to their audience. The log in credentials will be protected by encryption. The communication between clients and the servers is going to be handled by Internet; and the connection protocol will be HTTPS so that the transferred packets are encrypted as well. Malicious log in and sign in attempts are going to be rejected by extra precaution.

The users will develop circles of their own. Initially, the users will not have the permission to access every detail of other users. In order to access more data about other users, they should be added into circles as the data allowed for others tend to be of a greater level with the familiarity and a stronger relationship. On the other hand, the users have the right to block other users and blocked users cannot see the profiles of the users that blocked them.

3.6 Global Software Control

Each user will create his own account by himself via user interface using his e-mail address and credentials. After they create an account, they will be ready to use Touravel's infrastructure. Once a user successfully registered, he/she will be able to login using his/her credentials.

Global control of the mobile application will be client based and will be shaped according to responses comes from server. When a user navigate to a screen, it will make a request to server. Server will response to client afterwards. According to response, UI Manager will populate user interface components.

User's responses from mobile application will be in JSON form and submitted to server via HTTP based RESTful web services. Before making any requests, it needs to be authenticated via login functionality. After authentication is approved, client will be able to communicate to server in any need.

3.7 Boundary Conditions

Initialization of IOS/Android Client

- User selects Touravel icon on the menu of the IOS/Android device.
- If user has not been logged in before, she will be shown a login screen which requests username/password pair. She will have three attempts in an hour. After three unsuccessful attempt, she will have to wait for an hour to use login functionality.
- If user successfully logs in, UI screens will be shown to her according to responses from server.

Termination of IOS/Android Client

- If user closes the application by pushing home button on the phone, application will run in the background but never be logged out. After clicking application's icon on the phone's menu, user will continue where she left in the first place.
- Termination of the IOS/Android Client may be succeeded by logout button under left navigation drawer of the application. It will send a request that specifies logout request in order to inform server that is client will be logged off.

Failure of IOS/Android Client

- If IOS/Android client fails to complete an action unexpectedly (out of the desired actions per se) authentication of the client will be taken and client will log in again in future attempts.

4. Subsystem Services

Server Decomposition

Server consists of web services, route generator, social relationship manager, key-value cache manager, database manager, and key value store.

- **Web Services** are responsible of gathering web requests from clients and sending web responses back to clients.

- **Route Generator** is responsible of generating new routes as requests from users. It has access to other people's routes from database.

- **Social Relationship Manager** is responsible of social relations between users such as mentions and private messages.

- **Key-Value Cache Manager** is responsible of handling key-value pairs for user. It will work under web services. It will keep track of mostly accessed key value pairs and store them in Key-Value Store.

- **Database Manager** handles all the database queries. Database manager is responsible for storage and retrieval of the data.

Client Decomposition

Client consists of business logic, information tracker, communication manager, and UI Manager.

- **Information Tracker** responsible gathering data from user's device. His/her whereabouts, route preferences, and social relationships. Information Tracker turns raw data into meaningful user actions for Business Logic to use.

- **UI Manager** handles user interface components. There will be couple of screens that is available to user.

- **Communication Manager** is responsible for communications with server. It will send requests and translates responses for Business Logic to use.

- **Business Logic** will run in the backend of the client. It will be responsible for organizing Information Tracker, UI Manager, and Communication Manager.

5. Glossary

Web Service: A Web service is a method of communication between two electronic devices over a network. It is a software function provided at a network address over the Web with the service always on as in the concept of utility computing. The W3C defines a Web service generally as a software system designed to support interoperable machine-to-machine interaction over a network. [4]

HTTP: The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. [5]

JSON: JSON or JavaScript Object Notation, is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is used primarily to transmit data between a server and web application, as an alternative to XML. [6]

MongoDB: MongoDB (from "humongous") is an open-source document database, and the leading NoSQL database. Written in C++. [7]

Android: Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. [8]

IOS: iOS (previously iPhone OS) is a mobile operating system developed by Apple Inc. and distributed exclusively for Apple hardware. It is the operating system that powers many of the company's iDevices. [9]

6. References

[1] C. M. Technologies, "Chronos - Find Your Time," [Online]. Available: http://www.getchronos.com/. [Accessed 20 December 2014].

[2] M. M. Application, "Moves - Activity Diary for IOS and Android," [Online]. Available: https://www.moves-app.com/. [Accessed 20 December 2014].

[3] Rove, "Effortlessly remember every single day of your life.," [Online]. Available: http://roveapp.com/. [Accessed 20 December 2014].

[4] Wikimedia, "Web service - Wikipedia, the free encyclopedia," 23 December 2014 . [Online]. Available: http://en.wikipedia.org/wiki/Web_service. [Accessed 25 December 2014].

[5] Wikimedia, "Hypertext Transfer Protocol - Wikipedia, the free encyclopedia," 27 December 2014. [Online]. Available: http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol. [Accessed 29 December 2014].

[6] Wikimedia, "JSON - Wikipedia, the free encyclopedia," 25 December 2014 . [Online]. Available: http://en.wikipedia.org/wiki/JSON. [Accessed 27 December 2014].

[7] I. MondoDB, "MongoDB," 2013. [Online]. Available: http://www.mongodb.org/. [Accessed 25 December 2014].

[8] Wikimedia, "Android (operating system) - Wikipedia, the free encyclopedia," 28 December 2014 . [Online]. Available: http://en.wikipedia.org/wiki/Android_(operating_system). [Accessed 29 December 2014].

[9] Wikimedia, "iOS - Wikipedia, the free encyclopedia," 26 December 2014 . [Online]. Available: http://en.wikipedia.org/wiki/IOS. [Accessed 27 December 2014].