SWE 574 Software Development as a Team, Fall 2014
Instructor: Suzan Uskudarlı

Web Based Living History Application
Requirements Specifications Document

07.11.2014
Revision 4.0

By: Eser Gökçe Karaca
Student Id: 2013719078

Revision	Date	Explanation
1.0	10.10.2014	Initial Requirements
1.1	20.10.2014	Modifications on section 2.1.1 Modifications on section 3 Modifications on section 4.1
2.0	26.10.2014	Modifications on sections 2.1.1, 2.2, 3 and 4.1
3.0	01.11.2014	Modifications on sections 2.1.1, 2.2, 2.3.4, 3 and 4.1
4.0	07.11.2014	Modifications on sections 2.1, 2.2, 3 and 4

Table of Contents

1. Introduction

This software specifications document is a guideline for designers and developers regarding the functions served by the system. The purpose of this software project is to develop a web based system regarding people's memories using the programming language Java, and MySQL environment, to do mainly the following:

- Users should be able to describe a location and time based memory. The descriptions will include a place (or several places if appropriate), the time, and the exact description of the memory.
- Memories should be able to be located via searching and browsing. Search by location, time and person should be supported by the application.
- Users should be able to make comments on the memories.

The Web based system functions and user interface details are given in the following sections of this document.

2. Requirements List

2.1 Functional Requirements

2.1.1 Functional Requirements for User
There are 4 groups of functional requirements for the user:

a) *User Management Requirements*

No	Use Cases(s)	Requirement
1.	CreateAccount Use Case	The system shall create an account with specific attirubutes.
2.	LogIn Use Case	The system shall allow the user to login with his/her credentials (his/her username and password)
3.	EditInformation Use Case	The user shall be able to modify his/her registration information.

b) Searching and Browsing Requirements

No	Use Cases(s)	Requirement
1.	SearchMemory Use Case	The system shall enable the user to select one of the memories from the results of a search and view that memory.
2.	ListMemory Use Case	The system shall present the results of the search according to relevance.
3.	BrowseMemory Use Case	The system shall provide the necessary interface for browsing memories.

c) Memory Management Requirements

No	Use Cases(s)	Requirement
1.	CreateMemory Use Case	The user shall create a new memory by giving a location(s), a time (a set of temporal info) and a description of his/her memory. The description can either be a detailed one or a very short one. The user shall be able to use tags.

The system shall support the specifications of temporal instances in the form of day and/or month and year and (if exists) time of the day.
The system shall also support temporal duration.
If time is provided by the user, then temporal supports the following information:
"Year" information is required.
"Month" information is optional. (If year is specified, month can be provided.)
"Day" information is optional.   (If month is specified, day can be provided.)
"Time" information is optional.   (If day is specified, time can be provided.)
If time is provided by the user, then:
"Hour" information is required.
"Minute" information is optional.
"Second" information is not meaningful in the description of a memory.
The user may not enter any time or temporal duration as well.

2.	ViewAMemory Use Case	The user shall view a memory's details.

d) User Response Requirements

|		
1.	CommentMemory Use Case	This is the community feedback aspect of the application.

The user shall make comments on a specific memory. The user shall choose one of the predetermined feedbacks such as the emotions: "I like that", "T agree with your memory", "I remember that", etc.

2.2 Data Requirements

Package Diagram of "User" Class

The class "Ordinary User" that exist in the system is illustrated below.
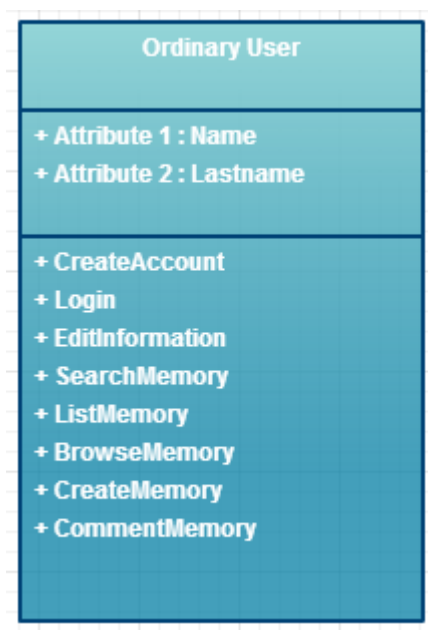
2.3 Non-Functional Requirements

2.3.1 Security

No security requirements have been provided by the customer yet.

2.3.2 Configuration

All relevant information should be able to be exported in RDF for interoperability purposes.
The application should have a web and an Android based mobile client.

2.3.4 Environment

Language: Object-Oriented Languages
Database: Scripting Query Language
The hosting will be provided by the customer.
Software methodology to be used can be agile.

3. Actors & Use Cases (Function Groups)

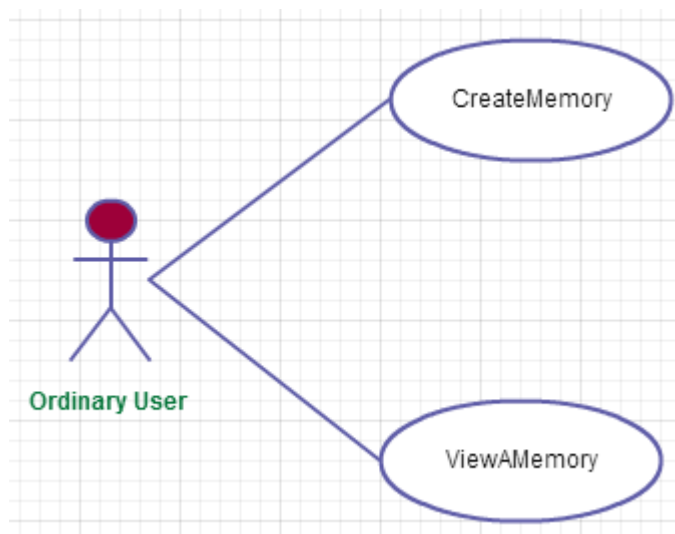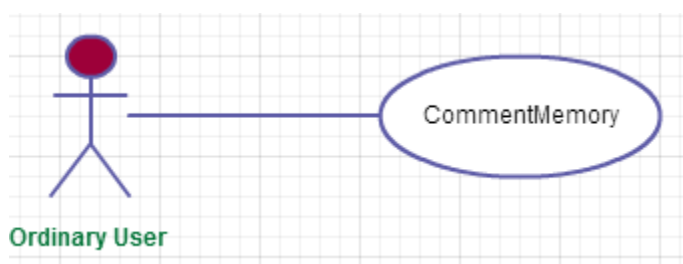| Actor	Description
Ordinary User	Any person that uses the system by entering new memories or viewing existing memories
Administrator	The administrator of the system that manages the web based database and has additional privileges compared to an ordinary user

User Management Use Cases
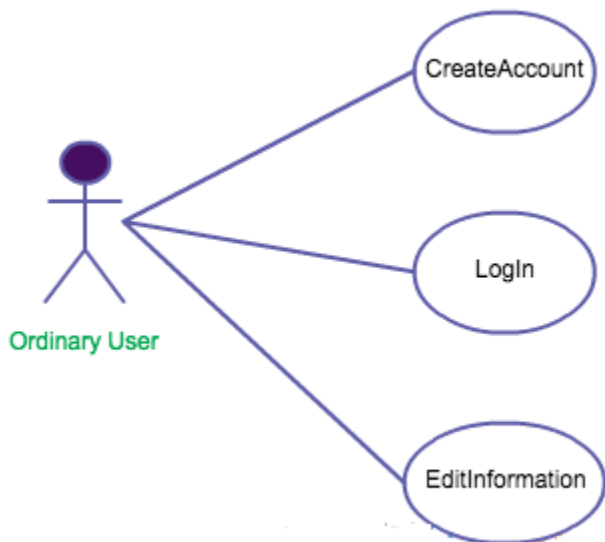
Searching and Browsing Use Cases
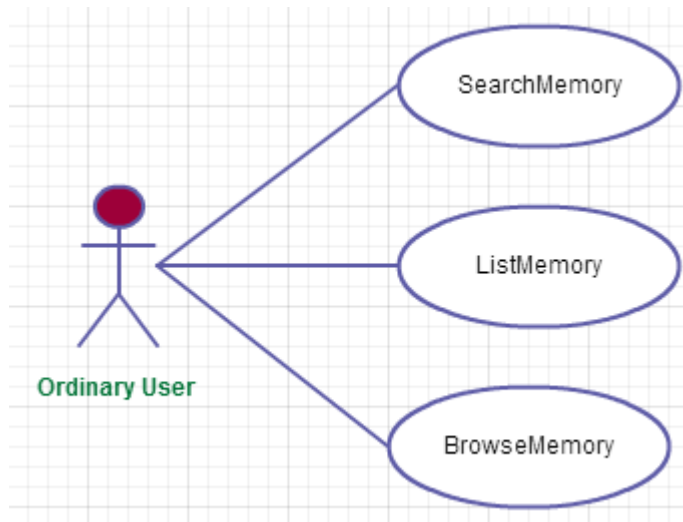
Memory Management Use Cases

User Response Use Cases

4. Function Definitions

4.1 Ordinary User Functions

No Use Case   Description

USER MANAGEMENT FUNCTIONS:

1. CreateAccount

    - This function lets any person to register her/himself to the system.
    - A person must fill out a form that contain some mandatory field such as e-mail info.
    - Reasonable lengths for all data fields and free format are assumed.
    - Button SUBMIT will cause an email with a web link to be sent to participant's email address with a proper message in email to activate
    - Account creation and login must be handled in a professional manner as much as possible

2. LogIn

    - This function enables users to identify themselves to the system
    - Unique e-mail address and password must be checked. If fields are matches with database, user must be redirected to the home page, else login form again.

3. EditInformation

    - Information that users state during the registration can be retrieved and listed by this function (except password).
    - While MY REGISTRATION INFO button displays necessary information, button UPDATE must be used to chance those info.
    - Button I FORGOT MY PASSWORD should mail the password of the user to his/her first entered email address (not the address entered on update screen). If user has forgotten his/her initial email address, there is no chance that he/she will retrieve the password.

SEARCHING & BROWSING FUNCTIONS:

4.   SearchMemory

- This function enables the users to search for memories with specific features.
- The memories can be located via searching and browsing.
- Advanced search by location, time and person are supported by the application.
- A map showing the location of the memory is presented using Openstreetmap at the time of viewing a memory.

5.   ListMemory

- This function displays the search results.
- Any user may invoke this function.

6.   BrowseMemory

- This function enables the user to browse through the results of the search..
- Any user may invoke this function.

MEMORY MANAGEMENT FUNCTIONS:

7.   ViewAMemory

- This function displays memories.
- Any user may invoke this function.
- The memory categorization is based on 2 aspects: time & location. Advanced search by location, time and person is supported.
- There are buttons to display previous and next memories for each location and to return to main menu.

8.   CreateMemory

- This function enables the user to describe a location and time based memory.
- Each memory can have one or more locations
- The user should enter a place, time and description for the memory.
- Several places can be entered for the same memory.
- The user is able to enter a memory description using multimedia elements (text, pictures, videos, sound recordings)
- There may be tags

MEMORY MANAGEMENT FUNCTIONS:

9. CommentMemory

> 🞦 This function enables the users to respond to or comment on the existing memories.

4.2 Administrator Functions

The administrator functions are not yet described by the customer.

5.   Glossary

Term	Description
Memory	In psychology, memory is the process in which information is encoded, stored, and retrieved.

6.   Initial Structure

Initial structure of the system depends on packages that include related use cases.

In this system, these are ordinary user and administrator subsystem packages, both of which access the memory application's database system.
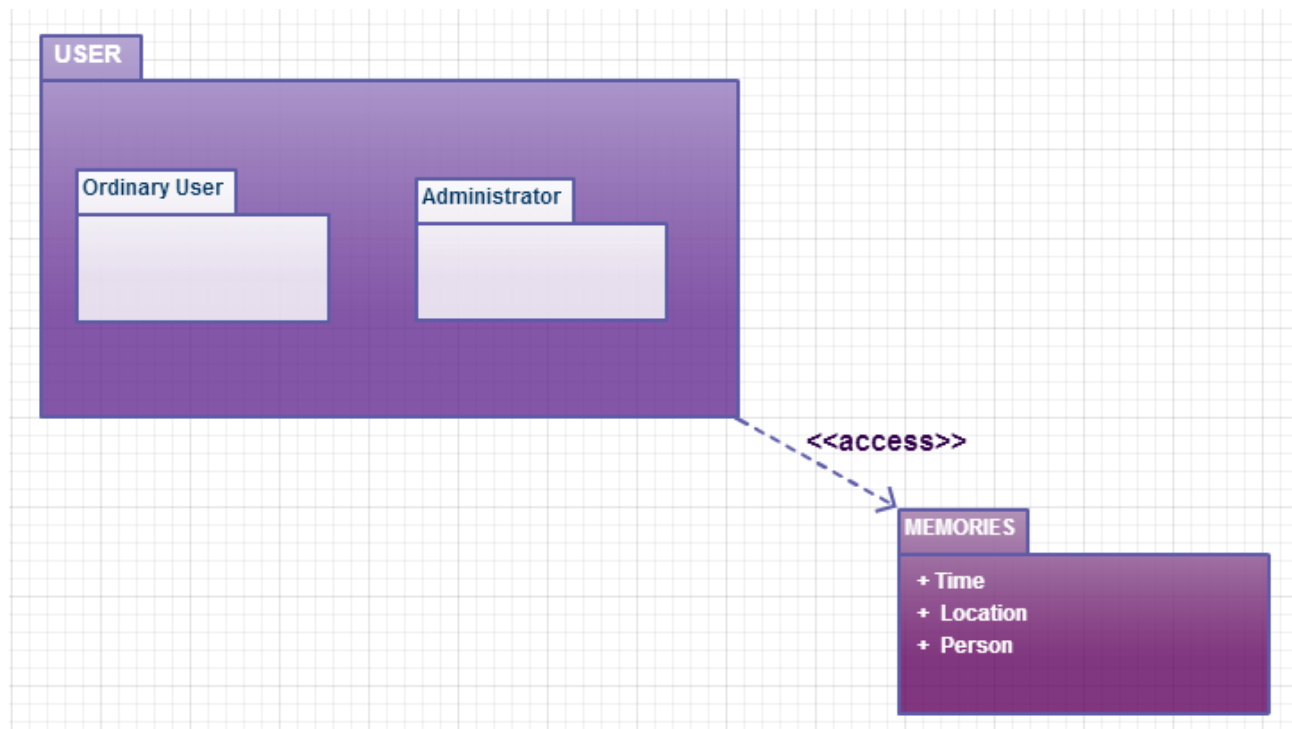
7.   References

Suzan Uskudarlı's Problem Definition document on http://moodle.cmpe.boun.edu.tr/mod/page/view.php?id=302 is used as a reference.