**SWE 574 Software Development As A Team, Fall 2015**

**Instructor: Suzan Üsküdarlı**

**Project**

**Requirements Specifications Document**

**19.10.2015**

**Revision 1.0**

**By Group 1**

Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Explanation** |
| 1.0 | 19.10.2015 | Initial requirements based on problem definition details and meetings with client |

Contents

[Revision History 2](#_Toc433129763)

[1. Introduction 3](#_Toc433129764)

[2. Requirements List 4](#_Toc433129765)

[3. Non-functional Requirements 7](#_Toc433129766)

[4. Overall Use Case Diagram(s) 7](#_Toc433129767)

[5. Use Case Descriptions 8](#_Toc433129768)

[6. Glossary 8](#_Toc433129769)

[7. References 9](#_Toc433129770)

1. Introduction

The purpose of this document is to determine the requirements of a system which facilitate the communication of a group of people interested in or working on the same topic. The application will be a system for **communities of practice** to obtain better mechanism for **communication**, **documentation** and **coordination of information**. Details of the application will be given in the following sections in this document.

1. Requirements List

The following table comprises a mapping between requirements list and use cases, it explores which use cases provide the functionality of each requirement.

| No. | Requirement | Use Case(s) |
| --- | --- | --- |
| 1.1 | Visitors can register with those information below:   1. Name, 2. Surname, 3. email, 4. password, 5. profile photo, 6. location, 7. hobbies, 8. short cv 9. role   A user may have one of the following user roles:   1. system admin 2. registered user 3. visitor 4. community admin (community owner) | Create User |
| 1.2 | Users can update their information except email.  Community owner role may be able to determine other member’s roles.  System admin can authorize any registered user to see reports on web based system | Update User |
| 1.3 | Users can search other users by user name or email | Search User |
| 2.1 | Only registered users will be able to create communities.  Creator of the community becomes community owner. | Create Community |
| 2.2 | Each community will have its own page containing questions and comments as posts in a chronological manner.  The user will be able to perform following operations from the community page  a. Schedule a meeting  b. See previous meetings list  c. Observe the records of a previous meeting  d. Upload a document or multimedia related to the community by giving tags  e. Send notification e-mails to the community members  These functionalities are described in separate use cases. | Show Community Details |
| 2.3 | All the information provided in the initial state of community creation will be available for editing afterwards. | Update Community |
| 2.4 | After the user logged in, the system will offer some communities that can be useful to the user. This list of communities will be created based on the communities the user has already joined and contributed. | Offer Community |
| 2.5 | All the communities that are previously created will be listed in a community listing page.  Community listing page will provide a functionality to search all the communities using keywords.  Users can search communities by tags  The results of the community page will be shown in a separate page containing links to community pages. The most accurate search result will be at the top and the least will be at the bottom.  Community listing page will contain sorting options for easy access. The options for the sort will be:  a. Sort by title  b. Sort by creation date  c. Sort by number of participant  Total number of communities will be shown at the community listing page.  Community listing page will provide a list of popular tags. These tags will be clickable and once they are clicked they will be redirected to communities having that tag. | Search & List Communities |
| 2.6 | Users can send request to join any community | Join Community |
| 2.7 | Community admins should be able to see the requests coming from users for joining the community. | List Community Join Requests |
| 2.8 | Community admins should be able to approve or deny the request coming from a user for joining the community. | Process Community Join Requests |
| 3.1 | Community members will be able to create meetings. The creator of the meeting should be the meeting manager.  The meeting should have the following attributes:   * Agenda - mandatory * Time (with timezone) - mandatory * Duration (in minutes) - mandatory * MeetingType (F2F, online, skype, etc…) - mandatory * Location – mandatory if meetingtype is F2F * Related documents (pdf) - optional * Meeting Manager (user) - mandatory * Participants (list of user) - mandatory * Minutes of Meeting(MOM) - optional   IRC integration should be available for online meetings.  Meeting manager will be able to upload pre-reading documents before the meeting.  MOM should be able to be created during the meeting.  Meeting attendants should only be the joined members of the community. | Create (Schedule) Meeting |
| 3.2 | Meeting manager will be able to postpone or cancel the meeting.  Any of the Meeting attributes should be editable afterwards. | Update Meeting |
| 3.3 | Community members should be able to see the list of meetings organized in that community. | List Community Meetings |
| 3.4 | Ongoing meetings should be listed in community details page | Show Ongoing Meetings |
| 3.5 | The manager of the meeting will be able to determine the roles of the Meeting participants. Roles will be note taker, and attendants. | Set Meeting Participant Roles |
| 3.6 | Meeting manager will be able to delegate managerial role to other members of the community. | Delegate Meeting Manager Role |
| 3.7 | Note takers will be able to create and upload meeting decision documents.  Minutes of Meeting document (MOM) should have the following attributes:   * Participants (may be different than the participants in Meeting obejct) – mandatory. * Actions (what, who, when) - mandatory * MOM Writer (user) - mandatory | Create MOM |
| 4.1 | Following reports should be shown in a dashboard style page:   * Total number of active and passive communities * Total number of active members of each community * Total number of active members * Total number of performed meetings * Total number of performed meetings of each community * …   Only system admin should be able to see the reports on the web based system  All reports can be only shown from web based system. | Show Reports |

Table 1

1. Non-functional Requirements
2. Environment:
   1. Linux or Windows Environment
   2. MySQL or similar DBMS
   3. C++ or C# or Java programming language
   4. Any development environment or framework
3. Logging: ?
4. Overall Use Case Diagram(s)
5. Use Case Descriptions

|  |  |  |
| --- | --- | --- |
| **No** | **Use Case** | **Description** |
| **1** | Create Community | Community Creation will be done by providing following information:  a. Title  b. Description  c. Tags  d. Initial member e-mails, names and their roles  e. List of outside people that can access community content  The creator of a community will automatically have the “community owner” role of the community.  System will send e-mail notifications after creation of the community. |

Table 2

1. Glossary

The Glossary list is shown in the table 1.

|  |  |
| --- | --- |
| **Term** | **Description** |
| Community | Communities are the spaces where people with an interest to a given topic meet. Inside communities various rich content can be found such as meeting notes, documents, papers, videos etc. |
| Visitor | Any unregistered user on system |
| User | Any registered user on the system |
| Admin | Administrator of the related system |
| Web Based System | The application which will be available on web. |
| Mobile Based System | The application which will be available on mobile devices. |

Table 3

1. References
2. <https://www.w3.org/community/>
3. Bennett-Object-oriented System Analysis and Design using UML 4thE