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

**Instructor: Suzan Üsküdarlı**

**Project BUCOMP**

**Requirements Specifications Document**

**26.10.2015**

**Revision 1.1**

**By Group 1**

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Explanation** |
| 1.0 | 19.10.2015 | Initial requirements based on problem definition details and meetings with client |
| 1.1 | 26.10.2015 | Use case diagram and use case details are added. |

# 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.

# 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 should be able to register to the system by creating their profile on the system. | Create User |
| **1.2** | Registered users should be able to update their profile information. | Update User |
| **1.3** | Registered users should be able to search other users by username or email address. | Search User |
| **1.4** | All registered users should be listed and user details should be displayed when clicked on the name. | Show/Update User Details |
| **2.1** | Registered users should be able to create communities. | Create Community |
| **2.2** | When clicked on the Community name, community details should be able to be viewed in a separate page. | Show Community Details |
| **2.3** | All the information provided in the initial state of community creation should 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. | Offer Community |
| **2.5** | All users should be able to search communities and list the result of any search. | Search & List Communities |
| **2.6** | Users should be able to 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 |
| **2.9** | Community admins should be able to upload resources. | Upload resources |
| **3.1** | Community members will be able to create meetings. | Create (Schedule) Meeting |
| **3.2** | Meeting manager should be able to update meeting information. | Update Meeting |
| **3.3** | Community members should be able to see the list of meetings organized in that community. | List Community Meetings |
| **3.4** | The manager of the meeting should be able to determine the roles of the Meeting participants. | Set Meeting Participant Roles |
| **3.5** | Meeting manager will be able to delegate managerial role to other members of the community. | Delegate Meeting Manager Role |
| **3.6** | Note takers will be able to create and upload meeting decision documents. | Create MOM |
| **4.1** | The system should be able to show required reports in a dashboard style page. | Show Reports |
| **5.1** | Community members should be able to upload media, document or write text for the posts section of the community. | Create Post |
| **5.2** | Community members should be able to write comment for the posts. | Comment Post |
| **6.1** | The system should display the user-specific items on landing page after logged-in. | Display User-Specific Items |

Table 1

# Non-functional Requirements

1. The system should be able to run on Linux or Windows environment.
2. The applications should be written in Java programming language and any development environment or framework is allowed.
3. The system should be 7/24 available.
4. Documentation should be provided as learned in SWE-573 course.
5. The mobile version of the app should be provided to the users.

# Overall Use Case Diagram(s)

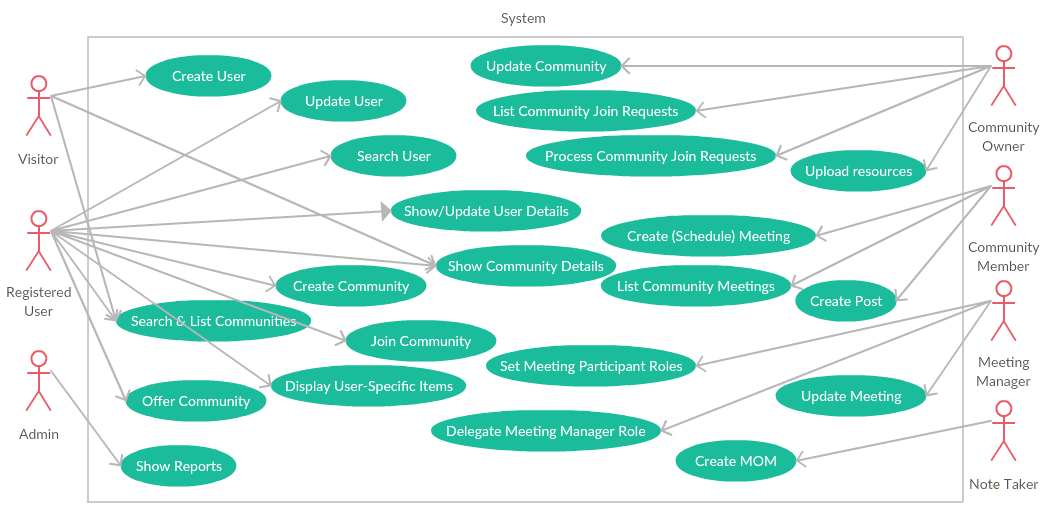


Figure 1

# Use Case Descriptions

|  |  |  |
| --- | --- | --- |
| **No.** | **Use Case** | **Description** |
| **1.1** | Create User | When a visitor clicks on the “register link” in any page, the registration form is displayed.Visitors should be able to register to the application by providing following information in the registration form:  a. name, mandatory  b. surname, mandatory  c. email, mandatory  d. password, mandatory  e. profile photo, optional  f. location, optional  g. hobbies (free text, max 255 char), optional  h. short cv (in pdf format), mandatory  i. role, mandatory (registered user by default)  A user may have one of the following user roles:   1. system admin 2. registered user 3. visitor 4. community owner 5. community member 6. meeting manager 7. note taker   After the visitor submits the registration form, the information is saved and login page will be displayed. |
| **1.2** | Update User | A form should be displayed when the “name” of the registered user on up-right corner is clicked. All user information except email should be editable in the form.  Only community owner role may be able to determine other member’s roles.  System admin can authorize any registered user’s role to see reports on web based system. |
| **1.3** | Search User | Registered users should be able to search other users by username or email address. The result of the search should be displayed as a list of users, and when clicked on the name, the details of that record should be displayed as a popup. |
| **1.4** | Show/Update User Details | All registered users should be able to be listed and user details should be displayed when clicked on the name. Only admin may be able to edit (users may edit information of themselves) their information. |
| **2.1** | Create Community | Registered users should be able to create communities.  Creator of the community becomes community owner.  Community Creation will be done by providing following information after clicking the “Create New Community” button in Communities home page:  a. Title  b. Description  c. Semantic Tags  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.  Only registered users will be able to create communities.  Creator of the community becomes community owner. |
| **2.2** | Show Community Details | When the user clicks on a Community Name, a new page should be opened and the details of the community should be viewed if the user is a member of that community, otherwise a summary (Community description, number of members, number of meetings) of the community should be displayed with a “Join Community” button.  The user will be able to perform following operations from the community page  2.2.1. Schedule a meeting  2.2.2. See previous meetings list  2.2.3. Observe the records of a previous meeting  2.2.4. Upload a document or multimedia related to the community by giving tags  2.2.5. Send notification e-mails to the community members  These functionalities are described in separate use cases.  The community details page will display following:  2.2.6. Community description  2.2.7. Community semantic tags  2.2.8. Questions and comments as posts in a chronological manner.  2.2.9. Ongoing meetings.  2.2.10. The names of community admins and members of the system. Members will be able to display the profile of these users by clicking on them.  2.2.11. Uploaded resources.  2.2.12. The history. This section will include the timestamp of such items:   1. Posts 2. Community creation 3. Meeting creation 4. Uploaded resources 5. Joined members |
| **2.3** | Update Community | All the information provided in the initial state of community creation will be available for editing afterwards. |
| **2.4** | Offer Community | After the user logged in, the system will offer some communities that can be useful to the user in communities home page. This list of communities will be created based on the communities the user has already joined and contributed. |
| **2.5** | Search & List Communities | All communities that are previously created will be listed in a community listing page (communities home page).  Community listing page will provide a functionality to search all the communities using keywords.  Users can search communities by tags  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.  The listing will have 15 entries, user should click on “see more…” button load more entries. See more button will load 15 more entries each time user clicks on that button. |
| **2.6** | Join Community | Users can join communities. The owner of the community determines the joining rule of the community whether it requires approval mechanism or not. If the community does not require approval mechanism user can join community. On the other hand, if it requires approval mechanism, user can send a request to join the community. |
| **2.7** | List Community Join Requests | Community admins should be able to see the requests coming from users for joining the community. The list should contain name, surname, email address, and location of the user. |
| **2.8** | Process Community Join Requests | Community admins should be able to approve or reject the request coming from a user for joining the community. |
| **2.9** | Upload Resources | Community admins should be able to upload resources such as media, video, and documents for the resources section on community detail page. Community admins should also tag the resource from the list or free text. |
| **3.1** | Create (Schedule) Meeting | 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  - Meeting Type (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. |
| **3.2** | Update Meeting | Any of the meeting attributes should be editable afterwards. |
| **3.3** | List Community Meetings | Community members should be able to see the list of meetings organized in that community. |
| **3.4** | Set Meeting Participant Roles | The manager of the meeting should be able to determine the roles of the meeting participants. Roles will be note taker, and attendants. |
| **3.5** | Delegate Meeting Manager Role | Meeting manager will be able to delegate managerial role to other members of the community. |
| **3.6** | Create MOM | 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 object) – mandatory.  - Actions (what, who, when) - mandatory  - MOM Writer (user) - mandatory |
| **4.1** | Show Reports | 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. |
| **5.1** | Create Post | On community detail page, community members will be able to create posts. Posts might be basic text, video, audio or an image. |
| **5.2** | Comment Post | On a post or a MOM, community members can write comment as text. |
| **6.1** | Display User-Specific Items | On landing page, after user logs into system, below sections are displayed to user:  6.1.1. News feed: If an action happens on a community that user is a member of, is displayed here. The actions are:  6.1.1.1. A new post  6.1.1.2. Comments entered into user’s post  6.1.1.3. New meeting creation  6.1.1.4. Ongoing and upcoming meetings  6.1.1.5. Uploaded resources  6.1.2. The list of communities that the user is member of. The listing will have 15 entries, user should click on “see more…” button load more entries. See more button will load 15 more entries each time user clicks on that button. |

Table 2

# Glossary

|  |  |
| --- | --- |
| **Term** | **Description** |
| Admins | Administrator of the related system |
| Agenda | List of items that are discussed or done during the meeting |
| 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. |
| Community Admin (Community Owner) | A registered user who creates the community and controls the community objects such as community participants, community attributes. |
| Dashboard | A page where the functions can be used by system admins. |
| IRC | Internet Relay .Basically a huge multi-user live chat facility |
| Log in | A function for registered users to enter the system with username and password |
| Meeting | A gathering of registered users which are selected by meeting creator for a specific reason or topic |
| Meeting Decision Documents | A document which contains decisions those made during the related meeting |
| Meeting information | Attributes of a meeting which are mandatory to create a meeting |
| Meeting manager | A registered user who creates the meeting and controls the meeting objects such as meeting participants, meeting attributes. |
| Meeting participants (Attendants) | Registered users who are invited and joined to a meeting |
| Mobile Based System | The application which will be available on  mobile devices. |
| Minutes Of Meeting (MOM) | The official notes kept during a meeting |
| Note takers | A role of meeting participants which has the responsibility of writing down the meeting actions during the meeting. |
| Pre-reading Documents | Abstract for related meeting |
| Profile | A page of a specific user where the user information and user related system function are shown. |
| Registered users | Any registered user on the system |
| Request | An application to be a member of a community or a meeting |
| Search | A function of the system which allows users to find related system objects via keywords or tags |
| Semantic Tag | Relating to the meaning of written words |
| System Admin | Controller user of the whole system |
| The System | The application which is available on web or mobile devices for facilitating the communication of a group of people interested in or working on the same topic |
| Visitors | Any unregistered user on the system |
| Web Based System | The application which will be available on web. |

Table 3

# References

[1] <https://www.w3.org/community/>

[2] Bennett-Object-oriented System Analysis and Design using UML 4thE