Group 6

QMUL Issues and Feedback System

ECS506U Software Engineering Group Project

Requirements Elicitation Report

1. List of Requirements

Functional Requirements

System

Number	Requirement	Type	Priority	Use Case
1	The system must have two types of users: Student and Admin	Functional	High	All
2	The system has 4 types of tickets: Service, ITL, EE Lab and EC	Functional	High	All
3	There are 4 types of admin: ITS, ITL, EE Lab and Student Support Officer (SSO)	Functional	High	All
4	System should store all tickets	Functional	High	All
5	All users must have a username/email	Functional (D)	High	All
6	All users must have a password	Functional (D)	High	All
7	All users must have contact details	Functional (D)	High	All
8	All users must be able to login with a username/email and password to seamlessly use the platform	Functional	High	All
9	System should be integrated with QMUL's single sign on	Functional	Medium	All
10	System should use QMUL's existing systems and user databases to validate users for single sign on	Functional	Medium	All

11	System should automatically log user out after a certain period of time	Functional	Medium	All
12	System should allow all admins to access user information	Functional	Medium	Access user information
13	System must allow user to contact Support or access SSPR to reset password	Functional	Medium	-
14	System should show history of all activity and actions done on a ticket by admin	Functional	Low	View tickets, Track tickets

Tickets

Number	Requirement	Type	Priority	Use Case
15	Students should be able to submit issue reports	Functional	High	Report issue
16	Students should be able to attach files to their issue report	Functional	Medium	Report issue
17	The system should be able to create tickets based on reported issue	Functional	High	Report issue
18	Tickets must have the user's ID of the user creating the ticket	Functional (D)	High	Report issue, Create EC application
19	Tickets must have the time and date of when created	Functional (D)	High	Report issue, Create EC application

20	Tickets must include description of the problem	Functional (D)	High	Report issue, Create EC application
21	Tickets must need their own unique ticket ID	Functional (D)	High	Report issue, Create EC application, View tickets
22	System must allow tickets to be assigned priority (high, medium, low)	Functional (D)	High	View tickets
23	Tickets must include the status of the ticket (ongoing, resolved, etc.)	Functional (D)	High	Track tickets, View tickets
24	Students should be able to view all of their created tickets	Functional	High	Track tickets
25	System must allow admins to view a list of all tickets assigned to them	Functional	High	View tickets
26	System should allow tickets to be sorted by priority	Functional	Medium	View tickets
27	System should allow tickets to be sorted by date (ascending and descending)	Functional	Medium	View tickets
28	The system should assign tickets to the appropriate admin based on the type of issue	Functional	High	Report issue, Create EC application
29	System should notify an admin when a ticket has been assigned to them	Functional	Medium	Report issue, Create EC application

30	System should allow students to update their tickets	Functional	Medium	Track tickets
31	System should notify students about updates to their tickets	Functional	Medium	Update ticket
32	System must allow admins to update and modify the status of their assigned tickets	Functional	High	Update ticket
33	System should allow tickets to be reassigned to a different admin	Functional	Medium	Reassign ticket
34	System must allow ticket priority to be changed	Functional	Medium	Update ticket
35	System should allow admins to add comments and feedback to tickets	Functional	Medium	Update ticket
36	System should allow admins to search for specific tickets based on keywords or ticket ID	Functional	Medium	View tickets
37	System must allow admin to remove tickets	Functional	Medium	Update ticket

EC

Number	Requirement	Type	Priority	Use Case
38	System must allow students to request ECs	Functional	High	Create EC application

39	Students should be able to select the module that the EC is for	Functional	High	Create EC application
40	System requires EC tickets to contain the type of claim (Self-certification, Standard)	Functional (D)	High	Create EC application
41	System must create an EC ticket after EC application is submitted	Functional	High	Create EC application

Non-Functional Requirements

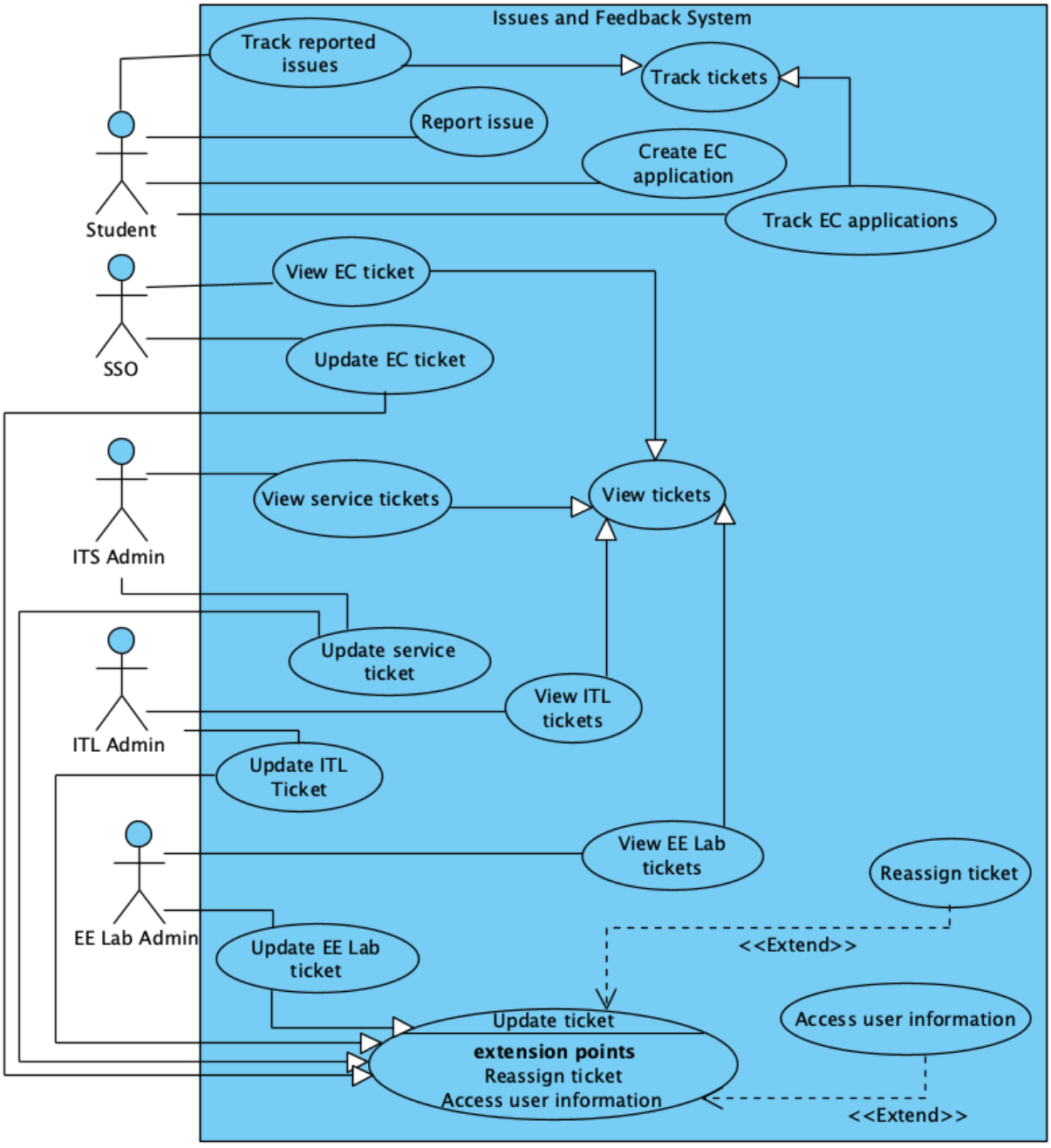
System

Number	Requirement	Type	Priority	Use Case
42	The system shall support a minimum of 500 concurrent users without any degradation in performance.	Non-Functional Performance	Medium	All
43	The system shall maintain an uptime of at least 99.9% to ensure continuous accessibility for users, minimising downtime for maintenance and updates.	Non-Functional Reliability	High	All
44	The system shall feature an intuitive user interface with clear navigation and easily understandable functionalities to facilitate usage by users with varying levels of technical expertise.	Non-Functional Usability	High	All

45	Users shall be able to access the system from any device with internet connectivity, ensuring compatibility and accessibility across desktop and mobile platforms.	Non-Functional Compatibility	High	All
46	Updates and patches shall be deployed seamlessly without disrupting system operations or requiring extended downtimed. Done in non school hours (21:00-23:00).	Non-Functional Maintainability	High	All
47	The system should be designed to scale easily to accommodate increasing numbers of users over time.	Non-Functional Scalability	Medium	All
48	The system shall be designed with responsive design principles, ensuring optimal display and functionality across various screen sizes, resolutions, and orientations, enhancing usability on mobile devices.	Non-Functional Usability	High	All
49	The system shall ensure a response time of less than 5 seconds for loading any page or performing any action, to provide a seamless user experience.	Non-Functional Performance	High	All
50	The system shall implement secure authentication mechanisms, including password hashing and encryption, to protect user accounts and sensitive information.	Non-Functional Security	High	-

51	The system shall be designed and developed with accessibility in mind, ensuring that all users, including those with disabilities, can navigate and interact with the interface effectively.	Non-Functional Accessibility	Medium	All

2. Use-Case Diagram



3. Use Case Description

1)

Name: Update ticket

Brief description:

Every admin can update tickets that are assigned to them. When the issue is being reported or an EC application is being created, a respective ticket will be created. When the issue ticket is created, the respective admin can resolve, update or track that ticket.

Actors:

- SSO
- ITL Admin
- ITS Admin
- EE Lab Admin

Preconditions:

- Admin needs to be logged in admin account
- Student needs to create an EC application or report an issue
- System needs to create corresponding ticket for EC or reported issue

Basic Flow:

1. System assigns a ticket to an admin.
2. The corresponding admin receives the issue that was reported, allowing them to take action.
3. Admin resolves the ticket.
4. Admin updates the ticket.
5. System changes the status of the ticket.

Alternate Flows:

<u>Admin needs additional information from student:</u>

3.1) The admin views ticket details and accesses user information.

3.2) Admin contacts the student via email, for the specific ticket being raised, to request additional information.

3.3) Flow continues from step 3.

<u>Admin cannot resolve the issue:</u>

3.1) The admin selects the option for passing the issue to a different admin.

3.2) Flow continues from step 1.

Post Conditions:

- Ticket's status was updated based on the admin action.
- Student gets notified when the status of the ticket has been changed.

2)

Name: Create EC application

Brief description:

Every student can complete an EC application. On completion of the EC application, the system creates an EC ticket. An SSO can then resolve, review and update the EC ticket. Students can also view and track the EC ticket they made.

Actors:

- Student
- SSO

Preconditions:

- User needs to have a student account
- User needs to be logged in

Basic Flow:

1. Student goes to the EC application section of the website.
2. Student selects the module that they would like to make an EC application for.
3. Student selects the specific task(s) that they would like an EC for.
4. Student provides a reason for not being able to complete the task(s) selected.
5. Student selects self-certification.
6. Student selects the date at which they will hand in the task(s) they are seeking the EC for.
7. Student submits the form.

Alternate Flows:

<u>Student is redirected to/selects standard claim</u>

5.1) Student has used up there three self-certifications and gets redirected to the standard claim (an automatic standard claim selection), or selects standard claim.

5.2) Student provides supporting evidence for the standard claim.

5.3) Flow continues from step 6.

Post Conditions:

- System creates EC ticket and it is assigned to the SSO.
- SSO gets notified when a ticket is assigned to them.

4. Risk Assessment

Risk	Likelihood	Severity	Impact	Preventative / Mitigating Actions
Resources and Team				
Poor time management (possibly due to high workload)	High	High	Unfinished product	Evaluate progress frequently and adjust project plan if necessary, using Kubernetes, a platform used for managing workloads
Missing member(s) due to health issues	High	Medium	Unfinished/ low quality product	Have frequent meetings and communication to make sure all members are aware of each other's progress, so that work can be picked up
Member(s) leaving the group (extenuating circumstances, disengagement)	Low	High	Delayed product	Consistent documentation of work so that there is a smooth transition when splitting up the work
Learning curves within the project group (eg. unfamiliarity with programming language being used)	Medium	Medium	Delayed product	Split up the sections that each team member will be required to learn, follow up with all members with thorough communication
Poor organisation	High	High	Unfinished product	Ensure every member is aware of the task that they have been assigned to and the deadline they must complete it by

Project Scope				
Gold-plating (adding features that are not part of the original scope)	Medium	High	Delayed product due to the need for correction, unsatisfied client	Requirements Scrubbing (removing complex requirements, ensure requirements are relevant to target market)
Project scope is not well-defined (could lead to scope creep)	High	High	Product is not relevant to what the client requires	Prioritise the requirements based on user needs, try to reduce any changes in scope
Requirements and Design				
Requirements are understood differently by members of the team as they are ambiguous	High	High	Incohesive product	Have frequent meetings to discuss requirements and make sure that the requirements are understood the same way by all team members
Design is not flexible/open to change	Medium	Medium	Client dissatisfaction	Have frequent meetings with client at each stage of process to see what changes they require
Design does not reflect the requirements	Medium	High	Client dissatisfaction	Constantly refer back to the requirements during the design process
Requirements are not fit for purpose	Medium	High	Product is not relevant	Involve stakeholders in requirements process to understand their needs
Incomplete requirements	Medium	High	Lost time, incomplete product	Use a structured approach, refer to use-cases and user stories continuously
Stakeholders				
Users have false expectations of the product being made	Medium	High	User requirements not being met	Ensure that the requirements are communicated clearly
Lack of stakeholder engagement	Low	High	Lost resources, potentially lost client	Maintain engagement by holding regular meetings with stakeholders and

				communicating ideas well
Client constantly changing requirements	Medium	High	Delayed product	Set strict deadlines and document requirements
External				
Unpredictable circumstances (eg. acts of nature) leading to loss of work and/or resources	Low	High	Lost work, extreme delays in product release	Have backups of work, save progress and documentation
Technological advancements (eg. new framework release)	Medium	Medium	Current framework does not consist of necessary tools that new framework does, lost time	Continue development on chosen platform to prevent delays