

School of Electrical Engineering and Computer Science
The University of Newcastle
SENG2050/SENG6050 Introduction to Web Engineering
Assignment 3 (35%) – Semester 1, 2021

Due Midnight Friday 4th June 2021
(Multiple milestone submissions)
IT Services Portal

1 Introduction

The university has an IT issue reporting system (<https://17000.newcastle.edu.au>). This system is available for staff and students to report issues they are having with various IT systems. Many organisations and institutions will use similar systems. In such systems, once an issue has been reported the system creates a workflow to assist in issue resolution. Issues go through many states in this workflow, such as: “new”, “in progress”, “waiting on third party”, “waiting on reporter”, “completed”, “not accepted”, and “resolved”. For example:

1. User Bob reports an issue he is having with connecting to the network.
2. The system marks this as a new issue.
3. IT staff can see this new issue. When they start work on it, the system will mark it as “in progress”.
4. At any time both the user and IT staff can add comments to the issue.
5. If at any time IT staff need to wait for a response from the user or a third party, they may mark the issue as “waiting on reporter” or “waiting on third party” respectively.
 - If the issue has been marked as “waiting on reporter” Bob would get a notification advising him to act on it.
6. Once IT staff have completed work on the issue they will mark it as such.
7. When Bob is notified that the issue is completed he will normally have a set time period (7 days) to:
 - Accept the change (marking the issue as “resolved”); or
 - Reject it – marking it as “not accepted” which notifies the IT staff again.

1.1 Your Task

It has been identified that IT staff are tending to similar issues repeatedly. This means that the IT staff are spending their time fixing the same issues, some of which could be fixed by the user on their own with the right knowledge. However, it is not a simple task for a user to locate the information they need to fix their IT issues. This is especially the case with sporadic issues.

Your task is to develop an IT incident reporting system with a built-in knowledge-base. IT staff can mark completed issues as “Knowledge-Base” in order to allow users to browse the knowledge-base before submitting an issue.

2 Requirements

For a group of 3, your system should meet the minimum requirements for a possible 65%. The remaining 35% will be assigned to additional requirements, of which you will need to make up a weight of 30.

For a group of 4, your system should meet the minimum requirements for a possible 50%. The remaining 50% will be allocated to additional requirements, of which you will need to make up a weight of 45.

i.e. If a group of 3 only implements the minimum requirements they will only be eligible for 65% of the final 35 marks available for this assignment.

Alternatively, if a group of 4 only implements the minimum requirements, they will only be eligible for 50% of the final 35 marks available for this assignment.

2.1 Minimum Requirements

The minimum requirements of the system are as follows:

2.1.1 User Management

There should be at least 2 different user roles in the system, users and IT staff. Users should see different elements in their views appropriate to their role.

You must include user registration/creation functionality. You should pre-fill the database with some users and provide a list of username-password-role triples.

Additionally, users should have a first name, surname, email, and contact number.

2.1.2 Issue Reporting

Users can report issues and IT staff can work to resolve them. Issues must have a state associated to them. Users and IT staff can change the state of an incident at any time (i.e. from “in progress” to “complete”). The following is how an issue’s state will change:

1. New – A user has reported an issue but IT staff have not started work yet.
2. In Progress – IT staff have started work, all interested parties can make comments.
3. Completed – IT staff have provided a solution.
4. Resolved – A user has accepted the solution (if the user rejects the solution, the issue will go back to “In Progress”).

Issues will be categorised as follows:

- Network
- Software
- Hardware
- Email
- Account

IT staff can view all issues in the system. They need to be able to sort issues by their status or issues should be grouped by their status.

Both IT staff and users can comment on an issue at any time before the issue is in its final state.

Users can only see issues that are in the Knowledge-Base or issues they have reported. They should not be able to see other user's issues.

Along with state and category an issue must also have a title, description, resolution details, comments (from both users and IT staff), date and time reported, and date and time resolved.

2.1.3 Knowledge-Base

IT staff may add an incident to the Knowledge-Base once it has been set to "Completed" or "Resolved".

A Knowledge-Base article is viewable by all users and should show at least the original issue's name, description, resolution details, and date and time resolved.

2.2 Additional Requirements

Following are some statements from the stakeholders of the system:

Note: Statements that have a stakeholder indicator (IT staff or User) have been made by those stakeholders. Statements without this indicator are general statements.

1. *"It would be good if the system could suggest related Knowledge-Base articles while I am reporting my issue"* (weight 10)
2. *"The current system works fine, I don't want a new system exposing my internal comments to users if I mark it as a Knowledge-Base article"* – IT staff. (weight 10)
3. *"Sometimes the language that IT staff use is complicated, maybe users should be able to add comments to the Knowledge-Base articles to help other users."* (weight 10)
4. *"The above suggestion is fine. However, I think we should be able to moderate this"* – IT staff. (weight 5)
5. *"Can we reply to comments on a Knowledge-Base article?"* (weight 5)

6. *"Can we reply to comments on an issue that we have access to?"* (weight 5)
7. *"It should be simple to link an open issue to an existing Knowledge-Base article"* – IT staff. (weight 5)
8. *"It would be nice if the new system had a section for us to advertise any planned or current maintenance"* – IT staff. (weight 5)
9. *"For a Knowledge-Base to work we need to be able to capture the data relating to an incident in a more meaningful manner. Currently we capture this data in a free-form textbox (Appendix 8.1). The suggested text is related to the category of incident but can be cleared or ignored by users"* – IT staff. (weight 20)
10. *"The categories are very broad and could congest the Knowledge-Base. Can we have some sub-categories as well (Appendix 8.2)"* – IT staff. (weight 20)
11. *"Sometimes our issues don't fit into a single category. Can we tag them with keywords as well?"* – User (weight 20)
12. *"The above suggestion sounds like a good idea. However, sometimes users have typos or ambiguous keywords. I think we should be able to change these before the issue becomes a Knowledge-Base article"* – IT staff. (weight 10)
13. *"I would like to stay anonymous if the issue I have reported is included in the Knowledge-Base."* – User. (weight 5)
14. *"We should be able to view Knowledge-Base articles sorted by their categories"* – User. (weight 5)
15. *"We should be able to sort the issues by the date that they were reported"* – IT staff. (weight 5)
16. *"It would be nice if we could have key-word search capabilities"* – User. (weight 20)
17. *"Not all incidents will make it to the Knowledge-Base right away. We need a useful way to search over the old incidents that aren't in the Knowledge-Base"* – IT staff. (weight 20)
18. *"It would be nice if we could attach photos and files to our incidents and that they could be displayed in the Knowledge-Base"* – User. (weight 25)
19. *"The above suggestion is a good idea but I think we should be able to remove unnecessary photos and files"* – IT staff. (weight 10)
20. *"I think we should be able to highlight the recommended solution, some of the proposed solutions from users might not be very appropriate"* IT staff. (weight 10)
21. *"Really, the system should just contain the "issue" and "solution(s)", not a big fan of a messy discussion board"* – User. (weight 10)
22. *"It would be nice if I could view the entire system on my mobile device as well"* – User (weight 15)
23. *"I hope my password is stored securely when signing up for this system"* – User (weight 5)

3 Web Technologies

You will implement the system using the web technologies taught in this course. At a minimum, you will use Servlets, JSPs and Beans, with JavaScript and CSS used where appropriate.

You may also use more advanced technologies such as STRUTS2, XSLT, or jQuery for dynamic JavaScript pages. However, their use will not attract more marks. Marks are awarded as per the specification, and not technical complexity.

You may use the IntelliJ IDE for this project. You may use the Gradle build tool for this course. If you use either of these technologies, you must submit your project as both an IntelliJ Gradle project, and exported .war file for the convenience of the markerx.

4 Databases

You will use a MSSQL database, as taught and used in this course. You may develop using a local copy Microsoft SQL Server, or use a

You must submit the complete SQL required to build the database (and add any initial data that your application requires).

Your application will use a JDBC named 'jdbc/itservices'. During development your data source will be defined in the application's context.xml file. Include a copy of your context.xml file in the root directory of your submission. BUT DO NOT SUBMIT IT IN THE META-INF directory. We will use a server-wide data source for marking. Please ensure you include your SQL database creation script.

5 User Logins

You will use tomcat's built-in users and roles functionality in your system. You will define a user realm in your context.xml, and submit it as part of your application.

You will include in your submission a list of user accounts.

6 Project Groups

This assignment is to be completed in groups of three or four students. Each team member will be given the opportunity to submit a peer review. Any claims of a team member not contributing will be treated seriously and require supporting evidence (emails, meeting minutes, etc.). If a clear case of a team member not contributing is discovered marks will be scaled accordingly. If you encounter such a case please let me know straight away.

7 Milestones

7.1 Groups Due – 6th May 2021 (0 marks)

Final group lists are to be emailed to me by this date at the latest. Any student not in a group by this date will be allocated a group by the 8th May 2021.

7.2 Draft Documentation and Plan Due – 14th May 2021 (10/35 marks)

Of the features you plan on implementing, you are required to submit the following drafts (using the UWE and UML standards where applicable <http://uwe.pst.ifi.lmu.de/index.html>):

1. Use case descriptions
2. Use case diagrams
3. Navigation models
4. Presentation models
5. Content model

Additionally, you will need to submit:

6. A Software Requirements Specification (see Blackboard for a template).
7. A development plan (which person is going to work on what, integration plan, timeline etc.), maximum one page.
8. Your group's lab time preference

I will create a schedule to meet with each group in the following week's labs in accordance with the group's preference. This time will be used to discuss the design, plan and any alterations.

7.3 Final Submission – 4th June 2021 (25/35 marks)

For this submission you are required to submit:

1. Your group's entire project that can be easily deployed in a Tomcat instance in the same way it is done in the labs
 - a. A webapps directory called "GroupX_FinalProject" where X, is your group name.
 - b. The main page should be shown when I enter the following URL in my browser:
http://localhost:8080/GroupX_FinalProject/
2. Context.xml with your database connection details
3. SQL to create and setup your database (including all initial data required by your application)
4. Any other config/setup files and scripts necessary to run your application

5. Readme.txt file describing the setup procedure
6. Revised documentation (as in milestone 2)
7. A document listing:
 - a. what minimum features are and are not implemented.
 - b. what additional features are implemented.
8. Group Assignment Cover Sheet

8 Appendix

8.1 Current details capture

I wish to report a problem occurring whilst using the internet.

My location (Building/Room):

My internet browser is (eg Internet Explorer v9 / Mozilla Firefox v12):

I am trying to connect to the following website:

I am able to access internal websites (Y/N):

I have tried using an alternate internet browser (Y/N):

I have tried restarting my computer (Y/N):

Problem description and error message:

8.2 Sub Categories

- Network
 - Can't connect
 - Speed
 - Constant dropouts
- Software
 - Slow to load
 - Won't load at all
- Hardware
 - Computer won't turn on
 - Computer "blue screens"
 - Disk drive
 - Peripherals

- Email
 - Can't send
 - Can't receive
 - SPAM/Phishing
- Account
 - Password reset
 - Wrong details

8.3 Marking Guidelines

The following is a general guide on how marks will be allocated to each part of the assignment:

Documentation: 10/35

(Split between milestone 2 (7 marks) and the revised versions in milestone 3 (3 marks))

- How complete is your requirements?
- Have you provided detailed analysis of the requirements in the form of use cases?
- Have you clearly shown how each requirement will be implemented in your system design?
- Are your designed elements “traceable” back to the original requirements?
- Does your documentation match the reality of the implemented system?

Implementation: 25/35

(Split between the functionality your group has implemented)

- Does your design follow the MVC (or similar) design pattern?
- Have you reduced/eliminated the use of scriptlets and java expressions/code in your JSPs?
- Does the system support multiple users?
- Is the HTML valid and well-formed?
- How efficient is your system?
- How usable is your system?
- Do you give the user sufficient feedback when operations have been performed?
- How secure is your system? Is there any way to break your system (SQL injection, XSS, modifying request parameters)?

- Does your submission contain all the file necessary to install and run your system? In particular, does it contain the scripts and data files needed to create and populate the database, and does it contain a readme file briefly explaining how to install the system?