Ticketing System Project Report

Ticketing System Project Report

Development of Information Systems Projects (DISP)

UFCFAF-30-3

Team 6

Team Members:

Ahmed Alharbi

Omar Almutairi

Mujeebur Alshamsi

Lewis Carroll

Muhammad Shayan Nagori

Liam Upstone-Smith

Project Goals:

Provide an easy-to-use ticketing system to expedite the organization's reporting and resolution of IT-related issues.    
   
Lower operating costs by managing IT support tickets internally and reducing dependency on outside providers.    
   
Using a centralised ticketing system, improve cooperation and communication between end users and the IT department.

Project Range:    
   
Create and implement an online ticketing system that includes features for creating, assigning, tracking, resolving, and receiving feedback from users.    
   
Add email integration and SLA management features.    
   
Improvements include time monitoring, survey capabilities for user input, and interaction with development issue tracking.

Acceptance Criteria

1. Ticket Creation and Submission:

- Users can raise tickets through two primary methods: a web portal and email.

- Web portal allows employees to log in and automatically populate relevant information.

- Email submission collects subject and body information to create tickets in the system.

2. Service Level Agreements (SLAs):

- SLAs are defined based on effect (High, Medium, Low) and priority (High, Medium, Low).

- SLAs determine response and resolution times for tickets.

- End users can select SLA metrics, but support staff can override them if necessary.

- Inactive tickets are automatically closed after two weeks to prevent backlogs.

3. Ticket Management:

- Tickets are initially read by the IT service desk manager for assignment and priority confirmation.

- Updates on tickets are communicated to end users via email for transparency.

- IT staff ensure sufficient information is gathered from end users to resolve issues.

- End users receive updates upon issue resolution and can confirm if problems are fixed.

- System reverts to information gathering stage if problems are not resolved initially.

4. Enhancements:

- Development issues are incorporated into the ticketing system, managed separately from support tickets.

- Development follows an agile framework with a focus on testing and user approval before deployment.

- Time tracking feature allows logging of time spent on tickets and development.

- Survey feature sends feedback requests to users after ticket completion for service improvement.

5. System Development and Deployment:

- BPMN is utilized to model the system operationally and strategically, documenting processes and identifying inefficiencies.

- System is developed using Java and a BPMN engine to create a demonstration system for proof of concept.

- Demonstration system is presented to the IT department for evaluation, approval, and deployment.

6. Continuous Improvement:

- System undergoes continuous monitoring and feedback gathering for further enhancements and optimizations.

- Regular updates and maintenance are performed to meet evolving business needs and technological advancements.

By meeting these acceptance criteria, the developed IT support ticketing system will effectively address the requirements outlined in the case study, providing efficient ticket management, transparency, and improvement opportunities for the IT department of the SME retailer.

Plan of Communication:  
  
Have weekly project meetings to go over challenges, developments, and future.   
  
Use Jira to manage tasks, issues, and progress. To keep everyone informed, update Jira often with task status updates and comments.   
  
Use WhatsApp as a platform for instant messaging to solve problems quickly and communicate with others.   
  
  
Assurance of Quality:  
  
Put in place comprehensive testing procedures.   
  
To guarantee the quality and maintainability of the code, conduct peer reviews and code inspections.   
  
  
  
Quick Reports.

First Sprint   
  
Achievements:   
  
Project plan in outline   
  
Problems:   
  
  
Tasks Scheduled for the Upcoming Sprint:   
  
Investigate several corporate architectures and decide one to adopt for the project.   
  
  
Second Sprint   
  
Achievements:   
  
System architecture research. The framework known as I\* was chosen.   
  
The Open Group Architecture Framework (TOGAF), Zachman Framework, and I\* Framework analyses have been uploaded to a GitLab repository.   
  
Problems:   
  
Tasks Scheduled for the Upcoming Sprint:   
  
  
Third Sprint   
  
Achievements:   
  
Problems:   
  
Tasks Scheduled for the Upcoming Sprint:

Fourth Sprint   
  
Achievements:   
  
Problems:   
  
Tasks Scheduled for the Upcoming Sprint:   
  
Fifth Sprint   
  
Achievements:   
  
Problems:   
  
Tasks Scheduled for the Upcoming Sprint:   
  
The report includes evidence of the project's quality and configuration management. (For instance, a thorough test strategy and proof of version control)   
  
Extensive Test Scheme   
  
The thorough test plan describes the testing methodology and techniques to guarantee the ticketing system's dependability, functionality, and efficiency. To check the system against predetermined criteria and standards, this approach comprises unit testing as well as system testing.   
  
The main goals are:   
  
Verify the ticketing system's features and functionality against the established specifications.

Determine and fix any flaws in the behaviour of the system.   
  
Verify that the system satisfies requirements for usability, security, and performance.   
  
  
1. Examining units   
  
- Goal: Maximise the usefulness of each part   
  
- Approach: Use Junit to create and run unit tests for every component.   
  
Pay close attention to assessing each module's data validity and error handling.   
  
  
2. Examining Systems   
  
- Goal: Assess the system to make sure it satisfies the needs for functionality, performance, and security.   
  
- Approach: Conduct comprehensive tests that encompass all aspects of the system, such as generating tickets, assigning them, resolving them, and gathering user input.   
  
- Test scenarios cover handling multiple ticket kinds, system answers, and various user responsibilities.

A strategy of validating the ticketing system against predetermined criteria and specifications is outlined in the comprehensive test plan. The strategy attempts to guarantee the system's performance, functionality, and dependability while providing an effective and user-friendly way for the company to manage IT support requests.   
  
agreed-upon project requirements   
  
The Ticketing System is intended to expedite the organization's reporting and resolution of IT-related problems. The features and functions required to satisfy end users' and the IT department's demands are listed in the criteria below.

Ticket Generation   
  
New tickets should be able to be created by users using an online portal or by emailing a specified email address.   
  
When tickets are created using the online portal, specific information ought to be automatically filled in by the system.   
  
Details of the Tickets   
  
When generating a ticket, users ought to be allowed to include a thorough description of the problem.   
  
Ticket Distribution   
  
The manager of the IT service desk should oversee allocating tickets to the right employees in accordance with their workload and area of expertise.   
  
Employees must have access to assigned tickets and any related information, such as urgency and priority.

Monitoring Tickets   
  
It should be possible for users to monitor the progress of their tickets and see if they are closed, open, or otherwise managed.   
  
When there are modifications to a user's ticket, such as when a ticket is assigned to a staff member or resolved, the system need to automatically notify the user.   
  
  
Resolution of Tickets   
  
When they work on a ticket, staff personnel ought to be allowed to update its status and indicate whether it has been resolved or is still open.   
  
When a ticket is resolved, employees should give a brief explanation of the steps they took and any solutions they produced.   
  
Interaction

Employees must be able to contact customers straight through the ticketing system, offering updates and, if needed, seeking more details.   
  
When there are revisions to their tickets, users must automatically be notified, guaranteeing openness and unambiguous communication all the way through the procedure.   
  
Mechanism of Feedback   
  
Users must be given the chance to comment on the resolution procedure and their general level of satisfaction with the assistance they got.   
  
After their problems are managed, consumers should receive a survey to offer input on the quality of service.   
  
Integration with Concerns of Development   
  
Together with support tickets, the system should be able to manage development-related issues and have different lists and locations for visibility.   
  
An agile framework should be used for development tickets, with an emphasis on user happiness and testing.

Time Monitoring   
  
Users should be able to track their time against tickets and development tasks in the system to receive insight into the productivity and burden of their department.   
  
These specifications set out the fundamental capabilities and attributes of the Ticketing System, guaranteeing an easy-to-use interface for users to report problems and effective workflow management for the IT division.   
  
agreed-upon project acceptance standards   
  
Collaboratively, accepted acceptance criteria were developed that specify the requirements that must be fulfilled for the project to be considered successful. The following is a summary of the agreed-upon criteria that served as our guidance and guarantee that we finished all that was required:   
  
Organised Scheme   
  
Workflow Organising

Pre/Post Examination with Junit   
  
Included should be the functionality that the case study required.   
  
Enter the details of a ticket request into the system.   
  
Change the status of the end user ticket.   
  
After a ticket is completed, close the request.   
  
Reactions to Surveys   
  
Assigning Assignments   
  
Right to cancel a ticket if inadequate information is provided   
  
Make a request for and use development ticket testing.   
  
Monitor log-in times for development concerns and tickets.   
  
These features are critical to our project's success and will be considered acceptable by the customer if they are met by the programme.   
  
A contribution matrix that has been agreed upon.

The responsibilities of each team member were outlined in an agreed-upon contribution matrix. Appendix 1's grid guaranteed responsibility and clarified each person's involvement. The contribution matrix was approved by the whole team, demonstrating shared ownership of the project's results, and honouring individual contributions.

We were able to track each task's progress in real time with Jira's powerful tracking features, which also provided information about the state of individual contributions. We used Jira to carefully define roles and duties, making sure that each team member's work matched their interests and areas of expertise. By classifying tasks according to their nature, complexity, and interdependencies, a methodical approach to project completion was made possible. Jira's frequent updates and status reports promoted efficient communication and let team members know how their colleagues were doing. In addition, we communicated any problems we were having via WhatsApp. We assessed each member's timely, high-quality, and impactful contributions, offering a comprehensive view of their participation in the project.

We reduced time spent on pointless activities and reduced the hazards associated with overworked team members by allocating resources dynamically.   
  
This approach has several advantages, including   
  
Jira's task tracking's transparency encouraged team members to perform their duties with diligence by creating a feeling of accountability.   
  
Because to Jira's superior resource management and job distribution, team productivity increased, and we were able to successfully complete project milestones.

reducing misunderstandings and obstacles to communication.   
  
In summary   
  
Effective project management techniques, strict quality and configuration management, adherence to predetermined requirements and acceptance criteria, and a well-defined contribution matrix all contributed to the project's successful conclusion. Communication, teamwork, and quality control were given top priority, and the result was a solution that satisfied the customer.   
  
  
Appendix 1: Matrix of Contributions

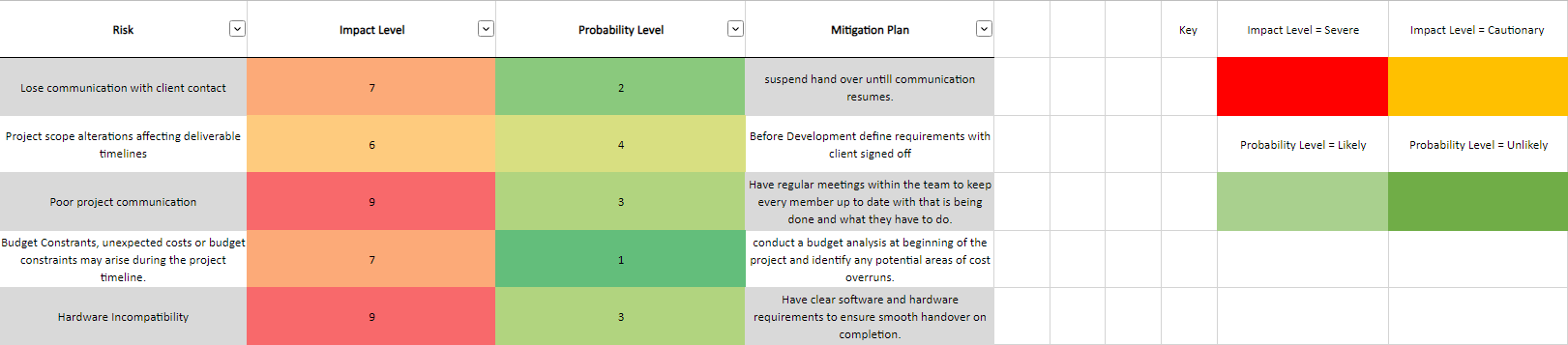
A white sheet with black text and numbers

Description automatically generated

The main points of the design rationale and development process of the Ticketing System are summarised in this paper, which highlights the value of cooperation, quality control, and efficient project management in attaining project success.

Risk Mitigation Plan

For us to identify any possible risks we may encounter during the development cycle; we have prepared a Risk Mitigation strategy (see Table 1). This enables us to plan and determine what will work best for all parties involved. Based on the likelihood that the risk will materialise and how it would affect the project's progress, the risks are ranked from 1 to 10. Every risk also includes a mitigation strategy, which the team has determined will be the next best step if the risk materialises into a genuine issue. For any risk, there are several ways for mitigating it, including risk acceptance, risk transfer, and risk avoidance. [What is risk reduction as of 2024?]. It is critical that we understand that not every risk will pose a danger to the project as a whole and that flexibility in response to a range of circumstances will be necessary to ensure the project's success.   
Table 1:



Test Plan

Testing is an important step in the development lifecycle, it is essential as to not release an unfinished product. The testing plan can be used as a step-by-step guide in the testing process. This means the plan should contain all the information the testers need for effective tests [What is a test plan? The complete Guide for writing a software Test Plan, 2024]. Although, the test plan will change throughout the project's lifespan, it is important to stay true to the agreed upon requirements of the project. The results from the test plan will be used to determine what needs to be focused on before the final deadline and the product is released. For test plan see Table 2.

Table 2

A test plan with text

Description automatically generated

Version Management   
  
GitHub served as our version control system for our project, allowing us to track and manage changes to our codebase. It functioned as a central repository from which every team member could get the most recent code. This made it possible for our team to work on the project while on the fly and update it as needed.

But we employed teams to develop any papers, including the report and the mitigation strategy. This enables real-time collaborative editing and viewing of the documents. This improved approach greatly increased our team's output and raised the project's overall quality.

