

Many Underpaid Millennials

Ticket Task



Project Report

Justin Valere : 816014152

Nicholas Smith: 816015916

Dhaniel Ramdath: 816015773

Table of Contents

[Introduction 3](#_Toc38032679)

[Problem Statement 3](#_Toc38032680)

[Goals/Aims of System 3](#_Toc38032681)

[Target User 3](#_Toc38032682)

[Requirements 4](#_Toc38032683)

[Functional Requirements: 4](#_Toc38032684)

[Non-functional Requirements: 5](#_Toc38032685)

[User Requirements: 5](#_Toc38032686)

[System Requirements: 5](#_Toc38032687)

[User Stories 6](#_Toc38032688)

[Story Case 1: Digicel Customer interaction with recommended solutions. 6](#_Toc38032689)

[Story Case 2: Agent at ISP firm takes call and submits ticket into system. 6](#_Toc38032690)

[Story Case 3: Agent's interaction with Ticket Task to aid in a customer’s ticket submitted through website. 6](#_Toc38032691)

[Use Cases 8](#_Toc38032692)

[Simple Uses Cases 8](#_Toc38032693)

[Use Case Diagram 9](#_Toc38032694)

[Use Case Ranking 9](#_Toc38032695)

[Expanded Use Cases 10](#_Toc38032696)

[Sequence Diagram 11](#_Toc38032697)

[Class Diagram 12](#_Toc38032698)

[Testing Plans 13](#_Toc38032699)

[Blackbox Testing: 13](#_Toc38032700)

[Acceptance testing 18](#_Toc38032701)

[Risk Management 19](#_Toc38032702)

[Cost Estimation 20](#_Toc38032703)

[Interfaces 21](#_Toc38032704)

[References 24](#_Toc38032705)

# Table of Figures

[Figure 1: Use Case Diagram 9](#_Toc38049123)

[Figure 2: Use Case Ranking 10](#_Toc38049124)

[Figure 3: Expanded Use Case 1 12](#_Toc38049125)

[Figure 4: Expanded Use Case 2 14](#_Toc38049126)

[Figure 5: Sequence Diagram 15](file:///C:\Users\dhani\OneDrive\SW%20Eng%20Group\Many%20Underpaid%20Millennials.docx#_Toc38049127)

[Figure 6: Class Diagram 16](#_Toc38049128)

[Figure 7: Table Showing Black Box Test for addUser() Method 17](#_Toc38049129)

[Figure 8: Table Showing Black Box Test for removeTicket() Method 18](#_Toc38049130)

[Figure 9: Table Showing Black Box Test for addToList() Method 19](#_Toc38049131)

[Figure 10: Table Showing Black Box Test for authenticate() Method 20](#_Toc38049132)

[Figure 11:Table Showing Black Box Test for selectPackage() Method 21](#_Toc38049133)

[Figure 12: Table Showing Risk Management 23](#_Toc38049134)

[Figure 13: Image Showing Cost Estimation Analysis 24](file:///C:\Users\dhani\OneDrive\SW%20Eng%20Group\Many%20Underpaid%20Millennials.docx#_Toc38049135)

[Figure 14: Login Page for General User 25](#_Toc38049136)

[Figure 15:Register Page for New Users 25](#_Toc38049137)

[Figure 16:Welcome Page for Employee 26](#_Toc38049138)

[Figure 17: Employee Add Ticket Page 26](#_Toc38049139)

[Figure 18: Technician's App Home screen 27](#_Toc38049140)

[Figure 19: Technician's App View Ticket 27](#_Toc38049141)

## Introduction

### Problem Statement

At present, many internet service providers have an automated user interface for customers to interact with and launch complaints if necessary. This process has long been observed to be frustrating to many customers, resulting in decline of customer satisfaction and subsequently, a loss of revenue. Considering this situation, we propose a ticket system which will encourage users to launch complaints in a quick and efficient manner, allowing customers to give feedback and for you to address common concerns among customers.

### Goals/Aims of System

The system would include an online interface which customers can access to submit their concerns which would be logged as ‘tickets’. These tickets would then be sorted according to category and logged in a database. The software would then compare the issue the customer is experiencing to other previously logged tickets in the database. The database stores all valid tickets ever submitted. Old solved tickets would also have their solutions and time taken to resolve stored in the database. When a new ticket is submitted, it is compared to pre-existing tickets and if a solution for the event is found, that solutions is presented to the technician handling the error. Time for the error to be handled is then allocated and a technician from the ISP’s staff is assigned the task.

### Target User

This system targets Internet Service Providers (ISPs) as the primary user-stakeholders, and further extends to their customers. Their customers submit their requests to the system, which organises the data for the ISPs.

## Requirements

### Functional Requirements:

• A user shall be able to navigate the ISP website to reach the complaint webpage

• A user shall be presented with a list of common problems of which to identify with

• A user should be directed to a package upgrade webpage if “Connection too slow” is selected

• A user shall be presented a comment box to explain in depth the issues occurring

• The system shall retrieve a customer address should “prolonged lack of service” be selected

• A dialog box should appear if “unsatisfactory customer service” is selected

• A comment box should appear if “other” is selected

• The system shall run the entered employee ID against all registered IDs and identify if a match is found

• A user shall be notified of the nearest branch location if “faulty equipment” is selected

• The system should automatically assign priorities to tickets.

• Priority levels should be incremented as time passes.

• Separate employee logins should be made

• Employee accounts can submit tickets with custom priority levels

• Employee accounts are presented with more detailed information when submitting tickets (e.g. Dispatch information)

• Technicians should be presented with a list of tasks assigned to themselves. ie. A technician can only see tickets he is designated

• Tickets presented to the technician should include all the client’s information

• Tickets presented to the technician should have suggested solutions based on similar tickets previously solved.

• A “Completed” option should be available to the technician to check off when the situation is dealt with

• A “Incomplete” option should be available to the technician to check off when the situation is incomplete and needs further work at a later time/date

• The incomplete option should increment the priority of the ticket and resubmit it to the queue

### Non-functional Requirements:

**Product requirement**

• The complaint webpage shall be available to all customers 24 hours a day, everyday

**External requirement**

• The system shall implement customer privacy provisions set ISO member data protection policy

• Customers who make false accusations against employees should be flagged

### User Requirements:

The ticket system shall allow for users to interact with a web page and select different options to request assistance.

### System Requirements:

• Each complaint made is logged and tailed by the system to produce a report every month.

• Every comment made in the comment box is saved and inspected by employees to determine if there is valid cause for concern or a previously undetected error.

• Upon employee log in their username & password is authenticated by the system before allowing access

## User Stories

### Story Case 1: Digicel Customer interaction with recommended solutions.

Nate is a customer of the Internet Service Provider (ISP) DIGICEL. He navigated the DIGICEL website and reached the customer complaint webpage. The webpage contains a list of common problems to choose a suitable complaint.

If “Connection too slow” is selected, the option redirects him to a webpage that displays package upgrades. Along with the upgrades he is given the option of a comment box for him to describe in depth the issues that occurred. Below the comment box is a submit and cancel button.

If “prolonged lack of service” is selected, their address is retrieved from the database and a technician is notified. The technician is then dispatched to the client's location.

If “unsatisfactory customer service” is selected, a dialog box will appear and customers are asked to identify the employee ID number. This will be logged and the person at fault will be notified and interviewed.

If “faulty equipment” is selected, customers are notified of the nearest branch location where they can exchange their defective equipment for a new model.

If “Other…” is selected, a comment box prompting for further details would appear.

### Story Case 2: Agent at ISP firm takes call and submits ticket into system.

Client of the ISP firm runs into an issue with his internet connection, the signal keeps dropping and has become highly inconvenient to the client. The client calls the firm and is connected to an agent. The agent hears the complaint and begins filing the ticket, choosing “Other…” option for category and specifies the situation in the comment box, with a custom priority level to the discretion of the agent. The system recommends that a technician be deployed to handle the situation. The agent then relays this information to the client and submits the ticket to the system, such that it would designate a technician to the issue and display the dispatch information to the agent, for the agent to relay to the customer.

### Story Case 3: Agent's interaction with Ticket Task to aid in a customer’s ticket submitted through website.

Agent double clicks the Ticket Task application on their work computer. The agent enters their username and password to log in to the Ticket Task. The agent double clicks the “queued tickets” tab. This displays a list of tasks assigned to that agent ranging from high priority (top of list) to low priority (bottom of list). The agent selects the ticket at the top of the list. This display a report of the client’s name, address, email address, phone number, current package deal, bill history, issue (if the client decided to enter an in-depth description) and the package upgrade selected. The client’s name will be highlighted and used as a link to retrieve their account from the database. The agent selects the modify account button shown at the bottom of the display window. This action allows the agent to change the base information of the client’s account. The agent is then given the option to change client’s name, email address, address, phone number and package deal. The agent selects package deal. A drop-down menu is displayed for the agent to view and select the appropriate package for the client. The agent selects the appropriate package then at the top right-hand corner of the account window. The client is then notified of the change and informed of the new monthly bill to be paid through email. The agent then checks a “Complete” option on the task to change its state and remove it from the queue.

## Use Cases

### Simple Uses Cases

User:

* Logins into system
* Creates Ticket

Employee:

* Views Tickets
* Updates Ticket

Technician:

* Updates Ticket
* Views Assigned Tickets

### Use Case Diagram

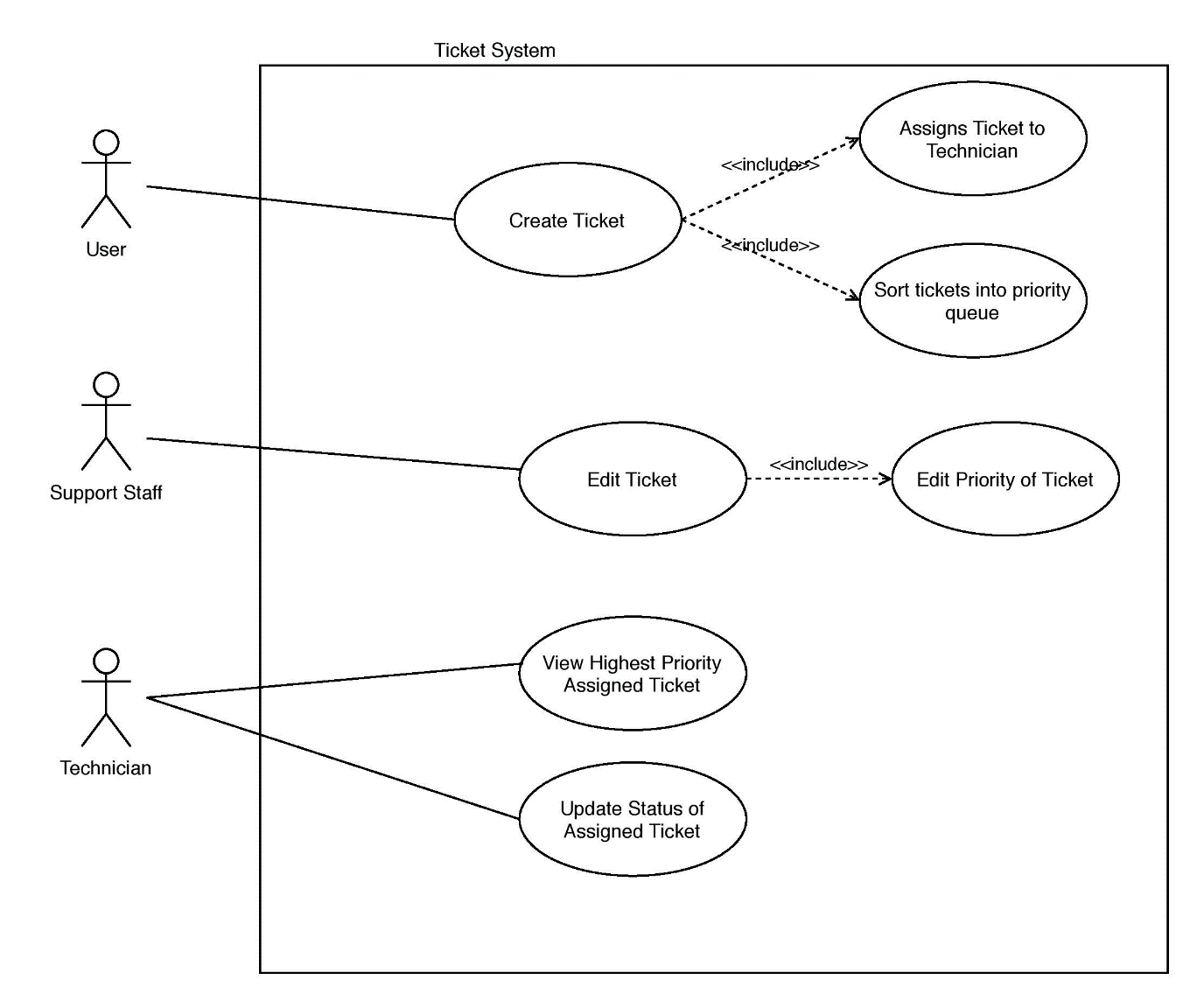


Figure : Use Case Diagram

### Use Case Ranking

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use-Case Name** | **Ranking Criteria, 1 - 5** | | | | | | **Total Score** | **Priority** | **Build Cycle** |
|  | **1** | **2** | **3** | **4** | **5** | **6** |  | | |
| Create ticket | 3 | 3 | 1 | 1 | 5 | 5 | 18 | Medium | 2 |
| Edit ticket | 1 | 5 | 4 | 1 | 5 | 3 | 19 | Medium | 2 |
| View highest priority assigned ticket | 1 | 1 | 1 | 1 | 1 | 1 | 6 | Low | 3 |
| Update status of assigned ticket | 1 | 3 | 3 | 1 | 4 | 5 | 17 | Medium | 2 |
| Assigns ticket to technician | 5 | 5 | 5 | 5 | 5 | 5 | 30 | High | 1 |
| Sort ticket into priority queue | 5 | 3 | 5 | 2 | 3 | 4 | 22 | High | 1 |

Figure : Use Case Ranking

### Expanded Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ticket Tasks System |  |  |
| Author(s): Nicholas Smith, Justin Valere, Dhaniel Ramdath | | Date: 20/03/2020 | |
|  |  | Version: 2 | |
|  |  |  |  |
| **Use-Case Name:** | Create Ticket |  | **Use-Case Type:** |
| **Use-Case ID:** | TTS-CT001 |  | **System Requirements** |
| **Priority:** | High |  |  |
| **Source:** | User Requirements |  |  |
| **Primary Business** | ISP Client |  |  |
| **Actor** |  |  |  |
| **Other Interested** | Support Staff |  |  |
| **Stakeholders:** |  |  |  |
| **Description:** | This use case describes the event of a client creating a complaint ticket | | |
|  | about a slow connection for Digicel Customer Care. The client logs in with | | |
|  | their valid username and password. Once the client information is | | |
|  | validated, the client can navigate to the webpage where they can select the | | |
|  | option of choosing a better package. The client is also provided with a | | |
|  | comment box to describe the issue in depth. The client submits the ticket | | |
|  | and receives an email notification of the ticket being successfully completed | | |
|  | and received. |  |  |
| **Precondition:** | The party must be a registered client of the ISP. | | |
| **Trigger:** | This use case is initiated when a user submits a ticket. | | |
| **Typical Course of** | |  |  | | --- | --- | | Actor Action | System Response | | Step 1: Customer navigates company website | Step 2: The system responds by validating the customer’s IP address to ensure he is a registered customer | | Step 3: Customer selects an issue | Step 3: If “Connection too slow” is selected, the option redirects him to a webpage that displays package upgrades | |  | Step 4: If “prolonged lack of service” is selected, their address is retrieved from the database and a technician is notified. The technician is then dispatched to the client's location | |  | Step 5: If “unsatisfactory customer service” is selected, a dialog box will appear and customers are asked to identify the employee ID number. This will be logged and the person at fault will be notified and interviewed. | |  | Step 6: If “faulty equipment” is selected, customers are notified of the nearest branch location where they can exchange their defective equipment for a new model | | Step 7: Customer calls the firm & is connected to an agent | Step 8: Agent acknowledges the complaint and fills out the ticket then refers to the system for a recommended course of action | | | |
| **Events:** |  |  |  |
| **Alternate Courses:** | Alt-Step 1: Call customer service hotline  Alt-Step 2: Describe issue in detail to service rep Alt-Step 3: Service rep confirms issue with customer Alt-Step 4: Service rep fills out ticket & submits to system for a recommended course of action | | |
| **Conclusion:** | Customers have a valid and efficient means of launching complaints to | | |
|  | appropriate members of staff |  |  |
| **Business Rules:** | Customers pay existing bill before a package upgrade is made available | | |
|  | Obscene language used by any customer will have that user flagged | | |
|  | Top priority is given to first time customers to ensure they return | | |
| **Implementation** | The ticket system to be implemented cannot work under conditions where | | |
| **Constraints and** | neither an internet connection nor phone line is available. This will render | | |
| **Specifications:** | the customer unable to launch complaints via tickets. Therefore a required | | |
|  | specification is either a working internet connection or phone line | | |
| **Assumptions:** | Every customer will not renew their package and need to be given an | | |
|  | incentive to do so |  |  |
|  | Not every customer is tech-savvy and can use the internet | | |
| **Open Issues:** | Power struggle among management. Ruling parties switch rapidly and | | |
|  | barely continue projects started by previous management | | |

Figure : Expanded Use Case 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ticket Tasks System |  |  |
| Author(s): Nicholas Smith, Justin Valere, Dhaniel Ramdath | | Date: 12/04/2020 | |
|  |  | Version: 1 | |
|  |  |  |  |
| **Use-Case Name:** | Assign Ticket to technician |  | **Use-Case Type:** |
| **Use-Case ID:** | TTS-CT002 |  | **System Requirements** |
| **Priority:** | High |  |  |
| **Source:** | User Requirements |  |  |
| **Primary Business** | ISP Client |  |  |
| **Actor** |  |  |  |
| **Other Interested** | Support Staff |  |  |
| **Stakeholders:** |  |  |  |
| **Description:** | This use case describes the events of the system assigning a ticket to a technician after evaluating the type and severity of the ticket. After the client has submitted a ticket with the relevant information. The system sends a confirmation email to the client. Keeps a record of email in the event of issues surrounding the ticket occurs. The system evaluates the ticket based on a priority ranking system. It then accesses a list of current technicians employed at the company. After it has selected the appropriate technician, it checks for the availability of the technician. | | |
|  |  | | |
|  |  | | |
|  |  |  |  |
| **Precondition:** | The client must select the submit button and has a valid ticket with required information. | | |
| **Trigger:** | This use case is initiated when a user selects the submit button on the webpage. | | |
| **Typical Course of** | |  |  | | --- | --- | | Actor Action | System Response | | Step 1: Customer enters required information in the provided fields. | Step 3: System sends a confirmation email to the client. | | Step 2: Customer selects the submit button on the webpage. | Step 4: System evaluates the ticket by analyzing the required fields entered by the client. | |  | Step 5: The system looks for keywords in ticket. The system has access to a list of keywords that provides a guideline of words. | |  | Step 6: It searches by keywords. | |  | Step 7: It accesses a list of employed available technicians. | |  | Step 8: The system assigns the ticket to a relevant technician keeping a record of the ticket ID and technician ID. | | | |
| **Events:** |  |  |  |
| **Alternate Courses:** | Alt-Step 1: System sends a notification to client declaring an incomplete ticket/insufficient information obtained.  Alt-Step 2: Outlines the information needed to complete the ticket. Alt-Step 3: Provide the client with a link to the webpage where they can complete the ticket. | | |
| **Conclusion:** | Technicians have a valid and efficient system of receiving tickets. | | |
|  |  |  |  |
| **Business Rules:** | Customers have no previous record of late payment for 2 consecutive months. | | |
|  | Obscene language used by any customer will have that user flagged | | |
|  | Top priority is given to first time customers to ensure they return | | |
| **Implementation** | The ticket system to be implemented cannot work under conditions where | | |
| **Constraints and** | neither an internet connection nor phone line is available. This will render | | |
| **Specifications:** | the customer unable to launch complaints via tickets. Therefore, a required | | |
|  | specification is either a working internet connection or phone line | | |
| **Assumptions:** | Some clients will not be satisfied with service provided by technician | | |
|  |  |  |  |
|  | Not every client is tech-savvy and can use the internet | | |

Figure : Expanded Use Case 2

## Sequence Diagram

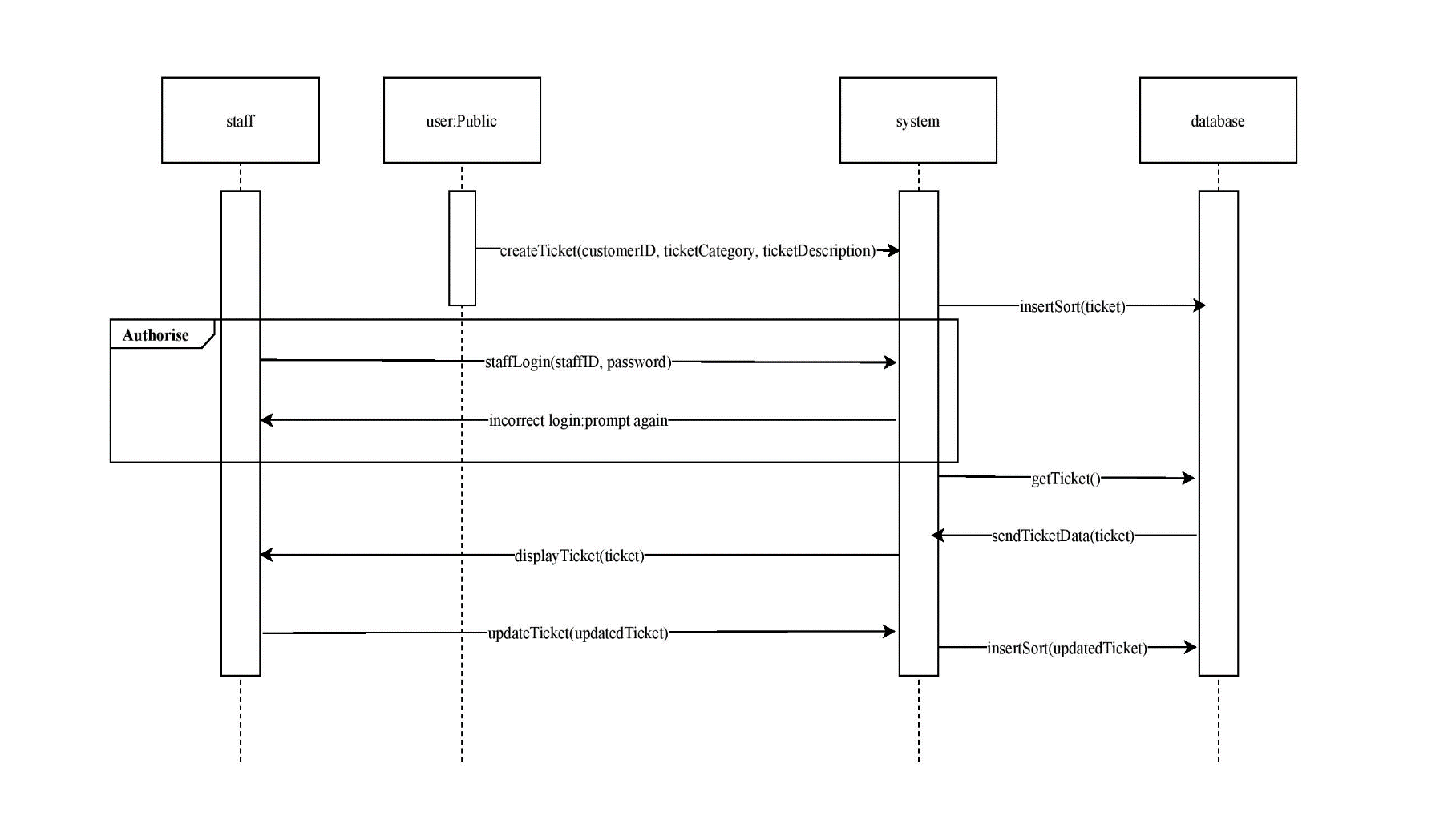


Figure 5: Sequence Diagram

## Class Diagram

A close up of text on a white background

Description automatically generated

Figure : Class Diagram

## Testing Plans

### Blackbox Testing:

#### Test 1

Method: addUser()

Description of Method: This method registers a new customer into the system and then allows the user to interact with customer representatives as well as launch tickets.

Criterion:

* User must not have a pre-existing account
* User has a valid ID
* User password must me 8 or more characters

Valid Class:

* Customer ID has never been used to create an account

Invalid Class:

* Customer ID is already in use by an existing account or not registered

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input type | Input | Expected output |
| 1 | Valid ID | User enters his previously unused ID to register | Notification to user that their account is created & prompted to choose a user name and password |
| 2 | Invalid  ID | User enters an ID used by another customer to register | Notification to user that ID entered is already in use & is prompted to enter a another ID |
| 3 | Invalid ID | User enters an ID unregistered by the system | Notification to the user that ID is not registered & is prompted to enter a valid ID |

Figure : Table Showing Black Box Test for addUser() Method

#### Test 2

Method: removeTicket()

Description of Method: This method allows an agent to remove tickets deemed invalid from the system as well as flag potential resource wasting users.

Criterion:

* Issue already identified from the same user
* Incomplete ticket information

Valid Class:

* Spam

Invalid Class:

* User has raised serious & original issue

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Input Type | Input | Expected output |
| 1 | Valid  Spam ticket | Agent repeatedly receives ticket already documented information | Agent removes ticket |
| 2 | Valid Incomplete ticket | Agent receives incomplete ticket information from user | Agent removes ticket |
| 3 | Invalid Complete & original ticket | Agent receives ticket that highlights an undocumented issue | Agent forwards ticket to be handled by developers |

Figure : Table Showing Black Box Test for removeTicket() Method

#### Test 3

Method: addToList()

Description: This method confirms the package the customer wishes to purchase the charges the customer’s account with the amount of credits equivalent to the cost of the package selected.

Criterion:

* Customer must be logged in
* Customer has selected a package to purchase
* Customer should have enough credits on their account to complete the purchase
* Device must be connected to the Internet

Valid Class:

* Package selected is less than or equal to the amount of credits on that user’s account

Invalid Class:

* Cost of package selected exceeds amount of credits user has on their account

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input type | Input | Expected output |
| 1 | Valid PackageCost<AmtOfCredits | Cost of package is less than the amount of credits the customer has on their account | Notification to customer that their package has been successfully purchased and their account has been reduced by the cost of the package |
| 2 | Invalid PackageCost>AmtOfCredits | Cost of package exceeds the amount of credits the customer has on their account | Notification to customer that they do not have sufficient funds to complete the transaction and is prompted to purchase more credits |
| 3 | Invalid PackageCost=0 | No package selected | Notification to customer that no package is selected |

Boundary: Amount of credits customer has on their account

Figure : Table Showing Black Box Test for addToList() Method

#### Test 4

Method: authenticate()

Description of Method: This method allows the user to authenticate themselves by entering a valid ID and password. The system will then check these credentials for validity.

Criterion:

* User must have a pre-existing account
* User has an valid ID
* User has a valid password

Valid Class:

* User enters a valid ID and password

Invalid Class:

* User does not enter a valid ID

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input type | Input | Expected output |
| 1 | Valid Verified ID & password | User enters an ID and password that matches one saved in the system database | Notification to customer that their login has been successful and allowed to move on to the next page |
| 2 | Invalid  Verified ID & inccorect password | User enters a valid ID but wrong password | Notification to customer that their password is incorrect and to recheck their spelling |
| 3 | Invalid Unverified ID | User enters an invalid ID | Notification to customer that no such ID exists |

Figure : Table Showing Black Box Test for authenticate() Method

#### 

Method: selectPackage()

Description: This method presents the user with a list of upgrade packages for him to choose from then calls the addToList() method once a choice is made

Criterion:

* Customer must be logged in
* Customer should have enough credits on their account to complete the purchase

Valid Class:

* User selects package with a higher connection speed than their current package

Invalid Class:

* User selects package with a lower connection speed than their current package

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Input type | Input | Expected output |
| 1 | Valid Package with higher connection speed than current package | User selects a package with higher connection speed that current package | Notification to customer that their package has been successfully chosen |
| 2 | Invalid PackageCost>AmtOfCredits | Cost of package exceeds the amount of credits the customer has on their account | Notification to customer that they do not have sufficient funds to complete the transaction and is prompted to purchase more credits |
| 3 | Invalid Package with lower connection speed than current package | User selects a package with lower connection speed that current package | Notification to customer that a package with lower connection speed that current package and asked to confirm whether they made the right decision |

Boundary: Amount of credits customer has on their account

Figure :Table Showing Black Box Test for selectPackage() Method

### Acceptance testing

#### Alpha testing

- for this part of the development cycle, a member of the development team will act as a typical customer attempting to use the system & actively engage with members of the organization who are responsible for assisting a given customer.

A member of the development team will interact with the proposed online system and fill out a ticket with relevant information then submit it. A member of our organization will be trained the expected amount of time it takes an employee to learn to use the new system and as act service rep and is expected to process the ticket and alert a technician if needed. Once the ticket is processed the service rep is expected to either provide a recommendation to the customer or provide a date on which the technician will arrive. The service rep must document any errors found while carrying out the required processes.

#### Beta testing

- once alpha testing has been successful a service rep from the target client will be trained the expected amount of time it takes an employee to learn to use the new system. A parent of one of our employees with little technical knowledge will be asked to participate in this stage of the development process. This parent will act as a dissatisfied customer and attempt to launch a complaint using the online system. The parent is expected to fill out a ticket using relevant information and submit it. Once the ticket is processed the service rep is expected to either provide a recommendation to the customer or provide a date on which the technician will arrive. When all required processes are completed both the parent and trained service rep will be asked to document any difficulty they encountered.

## Risk Management

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risk | Probability | Effects | Affect | Strategy category | Strategies to be employed |
| Personnel with critical skills unavailable | Moderate | Catastrophic | Project | Minimization | Employ people with overlapping skills |
| Personnel need extra time to learn unfamiliar tools | High | Serious | Project | Minimization | Create easy as-you-go guidelines to follow |
| The database used in the system cannot process as many transactions per second as expected | Low | Insignificant | Product | Minimization | Investigate the possibility of buying a higher performance database |
| Underestimated development time | Moderate | Catastrophic | Project | Contingency | Investigate use of a program generator |
| Key staff are ill at critical times in the project | Moderate | Serious | Business | Avoidance | Reorganize team so that there are more overlaps of work and people. |
| Technology change | Moderate | Serious | Business | Contingency | Investigate the possibility of using newer technology. |
| Product Competition | High | Serious | Business | Minimization | Invest in advertisements to publish way before the competition’s advertisements. |
| Requirements change | High | Serious | Project and Product | Minimization | Derive traceability information to asses requirements change impact. |
| Staff turnover | Moderate | Catastrophic | Project | Contingency | Have the old staff develop an easy guide on how to continue maintaining the product |
| Incomplete user functional requirement | High | Serious | Project and Product |  |  |

Figure : Table Showing Risk Management

## Cost Estimation

Figure 13: Image Showing Cost Estimation Analysis

For Cost estimation we decided to use a method based on Function Points to do our analysis. This gave us a rough estimate of 99 Unadjusted Function Points, which gave the above result in the chosen costing calculator above. It should be noted that this is subject to change based on the companies’ pre-existing logging and other related systems.

## Interfaces

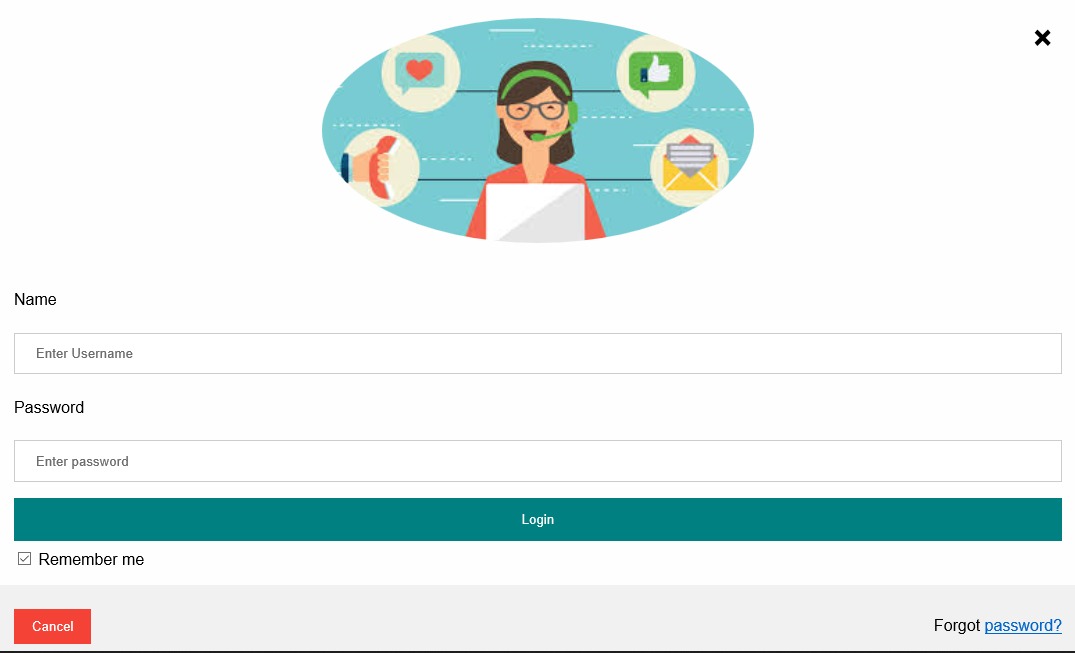


Figure : Login Page for General User

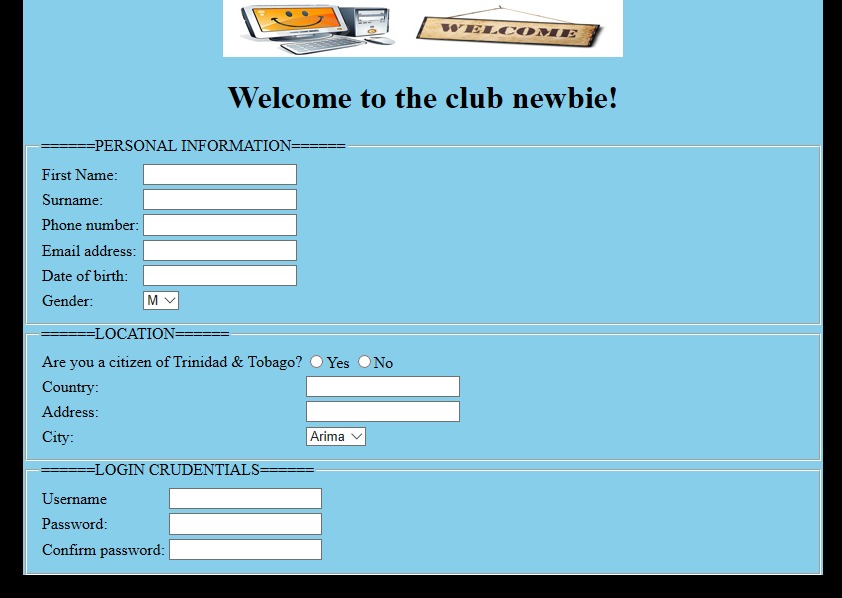


Figure :Register Page for New Users

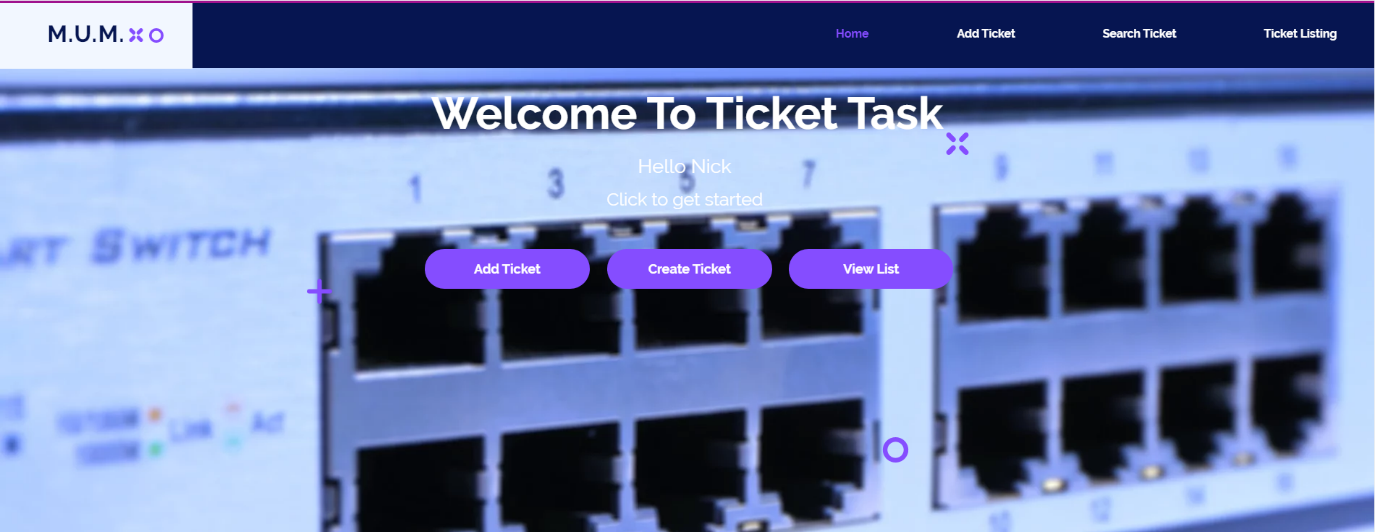


Figure :Welcome Page for Employee

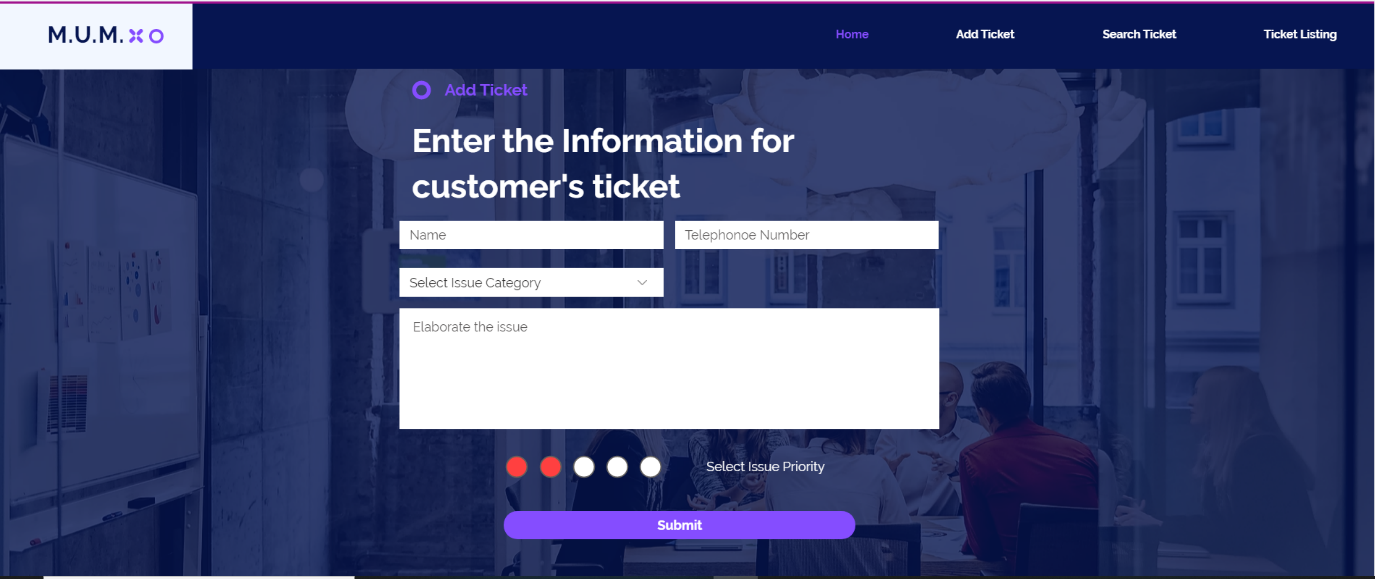


Figure : Employee Add Ticket Page

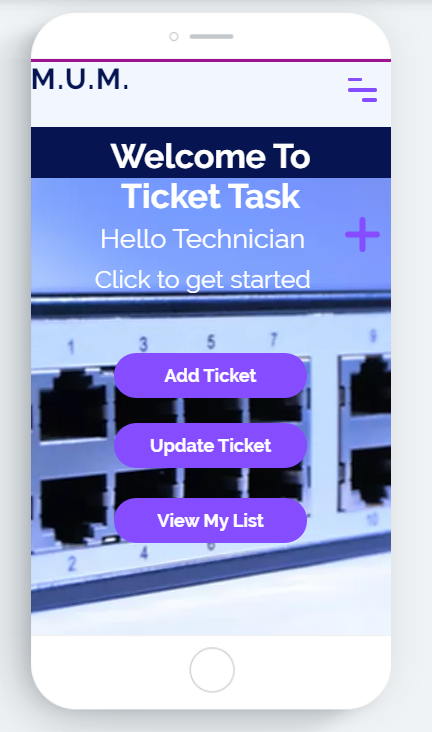


Figure : Technician's App Home screen

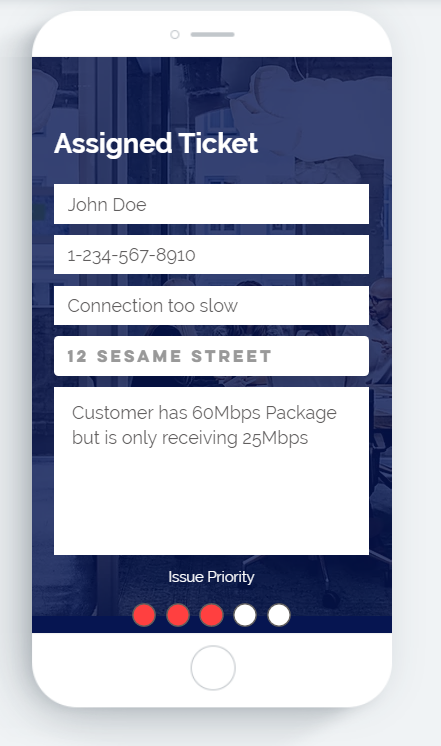


Figure : Technician's App View Ticket

## References

*Computer Standards & Interfaces*, 1992. Software engineering risk analysis and management. 14(2), p.180.

S. Mohammed, (2020). Use Case Examples and Guidelines. [Pdf file]. <https://github.com/firepenguindisopanda/Software-Engineering-Notes/blob/master/Use_case_Modelling/Use%20Case%20Guidelines%20and%20Examples.pdf>

*Software Engineering | Calculation of Function Point* GeeksForGeeks.com <https://www.geeksforgeeks.org/software-engineering-calculation-of-function-point-fp/>

S. Mohammed (2020). Class Diagram [Pdf file]. <https://github.com/firepenguindisopanda/Software-Engineering-Notes/tree/master/class_diagrams>

S. Mohammed (2020). Black Box testing example. [Pdf file]. <https://github.com/firepenguindisopanda/Software-Engineering-Notes/blob/master/Black%20Box%20Testing%20Example.pdf>

Ticket Task Github Repository Link: <https://github.com/firepenguindisopanda/Ticket-Tasks>