**Introduction**

The software to be developed is expected to be a web application that falls under the category of interactive transaction-based applications. It is planned to be a Ticketing System which is meant to be used by the customer support representatives and the customers. The system is to provide an online solution for the customers to bring their issues and concerns to the customer support people’s attention as well as for the customer support representatives to promptly address their customers issues. There will be two view portals of this web application. One portal is for the customers and the other is for the Back-Office users. It is expected to provide a service through which the customers will be able to generate tickets for their problems and issues using their online portal. The tickets created will be listed in that particular customer’ profile. As soon as a ticket is created, there will be a notification showing on the Customer Representative’s or Administrator’s Backoffice account. The created ticket will also be listed in the Administrator’s view under that customers name who initiated the ticket. Though the final deliverable will be an independent system, it is assumed to be able to get merged in an existing online Business Software platform.

**User Requirements Specification**

Functional Requirements:

As the user visits the main page of the ticketing system, it will land on the a page which provides a login form and also a signup. The user can either login if it is a returning user or signup if it is a new user.

1. If the user is new, he/she will click on the sign-up button. This will proceed the user to the registration form. The registration form will include necessary information to sign up. Once the user fills the form and clicks ‘Submit’, the user’s data will be stored and the user can proceed to login.

In case the user chooses to login,

1. the user will input the username and password and click on “Login” button. The system will authenticate and authorize the input and forwards the user based on his/her role to its corresponding portal/ profile page.
2. After the sign in is successful and the incoming was a customer, the customer will see his profile with tickets listed that he/she created in the past.
3. Moreover, there will be an option to create a new ticket by clicking on the button “Create New Ticket”. Once the button is clicked the system will direct the customer to a form asking to provide the following information: Ticket type (categorizes the issue the customer is facing to be directed to the right support group), Subject and Description of the problem encountered. Once all required fields are provided, the customer can click the “Submit Ticket” button to submit the form. The customer will then be redirected to its profile again and the newly created ticket can be seen in the ticket queue.

However, if the incoming user is a Backoffice user i.e. the administrator or a customer support representative, then

1. the user profile would list all the customers registered in the system.
2. The Backoffice user can go into each customer profile by clicking on the customer’s name and can see the tickets listed pertaining to that specific customer. The Backoffice user will have the option to update or delete the existing tickets as well as create new tickets on behalf of that customer.
3. The admin user will receive a notification whenever a customer creates a new ticket. The admin can also assign the ticket to customer support representative.
4. Additionally, the administrator role will also have the ability to create a new user account. It could be either a customer account or representatives account.

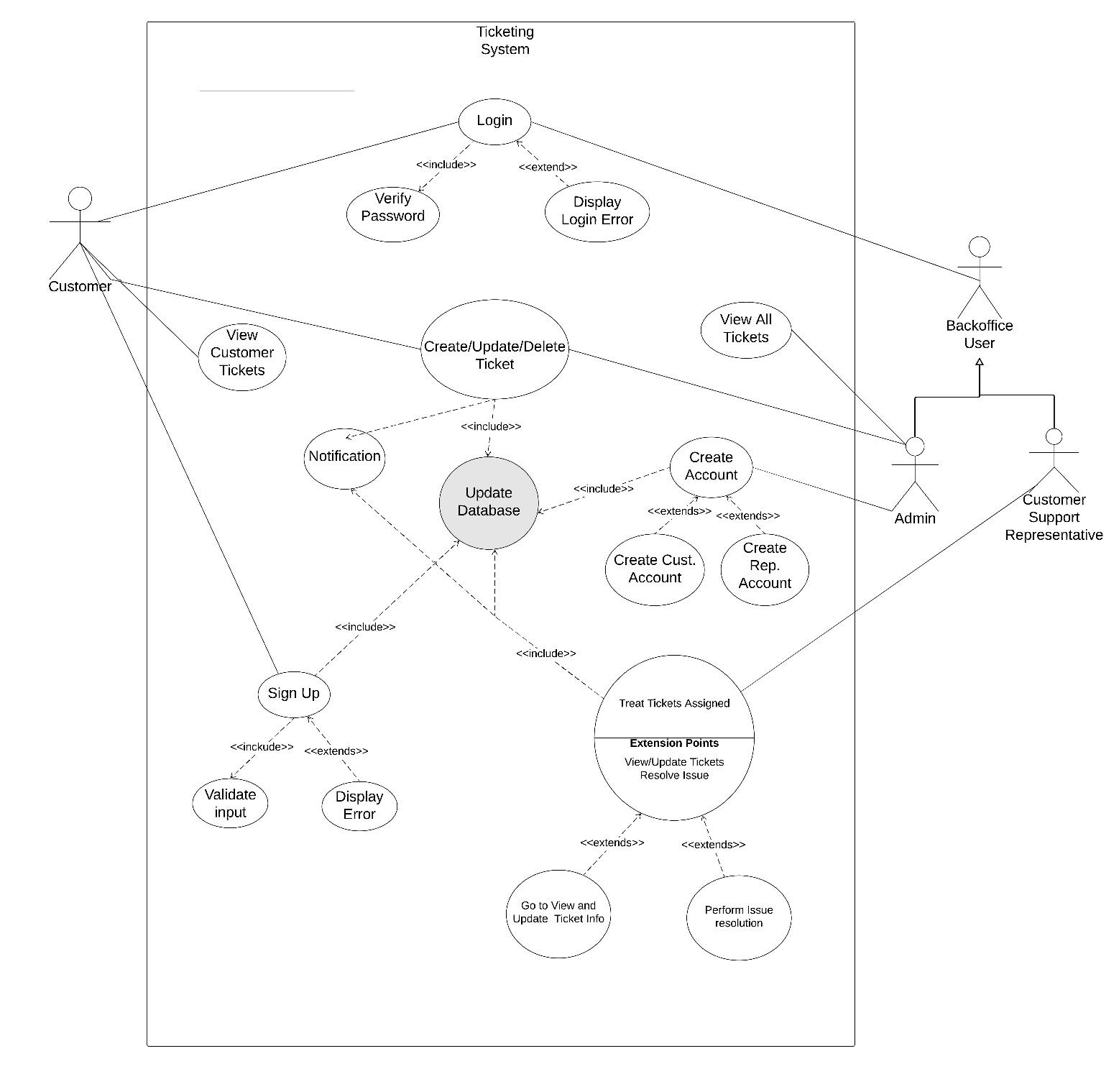


Figure 1: Use Case Diagram

[The above Diagram describes the use cases of the ticketing systems.]

Non-Functional Requirements:

Application would be responsive. In case of failure, timeout shouldn’t be more than 5 seconds. As customer creates the ticket, it should appear on customer’s profile immediately. As back-office user creates, updates, or deletes any ticket, it should, immediately, update the both customer’s and back-office user’s portal based on the operation performed. The response time should not be more than 2 seconds. The backend processing and business logic will be structured such that the server response is quick. The forms to fill and the functionality to create/update/ delete tickets and accounts will be simple and self-explanatory.

**System Requirements Specification**

Functional Requirements:

Before accessing customer’s portal, you must authenticate yourself with sign-in information. The sign-in page should include a form containing two input field for username and password. If you click on “Login”, it will check the database if customer exists with provided username and password. The password, in the database, must be decoded before comparing it with provided password through sign-in. If you are authenticated, you will be redirected to the customer's portal, otherwise it will stay in the same page and display an appropriate error message. Customer portal should include the list of tickets exist in the database for that particular customer if there is any. If you click on “Create New Ticket”, you will be forwarded to page asking information about creating ticket. The page should include a form containing the following input fields

* status (New,Feedback, Acknowledged, Assigned, Resolved or Closed),
* priority from (Low, Normal, High, Urgent)
* summary (Text of at most 100 chars)
* description (Text of at most 750 chars)

Customer are allowed to create and update their own ticket. Customer should not be able to see other customers’ tickets and information.

Back-office users must also authenticate themselves through the login information with username and password. After you click “Log In”, it should check database with the provided username and password. In addition, it should also check if user is a back-office user by checking the role in the database table. After you are authenticated as a back-office user, you will be redirected to the back-office user’s portal, otherwise it will stay in the same page and display an appropriate error message. Back-office users’ portal should list all customers as a link from database and if you click on any customer, you will be redirected to customer’s profile with all ticket listed created by that customer. Back-office users are allowed to create, update and delete the for customers. Any update is made to the ticket by back-office user will update the database as well as the customer’s portal displaying the updated ticket,

|  |  |
| --- | --- |
| Function | Create Account |
| Description | The admin will input information for the customer or representative’s account |
| Inputs | First Name, Last Name, Email |
| Source | The Admin |
| Outputs | A profile that can be used by either a customer or a rep. to create and manage tickets. |
| Destination | Database |
| Action | Type in the information and submit |
| Requires | The admin to input the proper information |
| Precondition | The admin must possess the necessary information |
| Postcondition | The customer or representative employee will have an account to work in the ticketing system. |

Figure 2: Create Account

[The above table describes the requirements of implementing the process of

creating an account.]

|  |  |
| --- | --- |
| Function | Create Ticket |
| Description | A new ticket record is created |
| Inputs | Customer, Ticket Description, status, priority, summary. |
| Source | The Admin, Customer |
| Outputs | A new ticket in queued into the customer profile and sorted based on the priority |
| Destination | Database |
| Action | Type in the information and submit |
| Requires | The admin to input the proper information, the customer info for which the ticket is being created. |
| Precondition | The user must have an issue for the ticket and also the ticket’s customer |
| Postcondition | A ticket is stored in the DB and queued in the list |

Figure 3: Create Ticket

[The above table describes the requirements of implementing the process of

creating ticket.]

|  |  |
| --- | --- |
| Function | Login |
| Description | The user will be logging in to its portal |
| Inputs | Username and Password |
| Source | The incoming user |
| Outputs | Entry into the user’s account |
| Destination | User’s Profile page |
| Action | Type in the credentials and click login |
| Requires | The user to input the valid credentials |
| Precondition | The user must know the valid username and password |
| Postcondition | none |

Figure 4: Login

[The above table describes the requirements of implementing the process of

login.]

**Platform Requirements Specification**

As for the user’s perspective, the user would require a PC which supports internet connectivity and any browser to access the system online. However, as far as the business personals and developers are concern, the system will be developed in a 64-bit machine with Windows operating system. The system will be developed as Maven Project using Java Spring MVC framework. The environment will be provided by IntelliJ idea IDE. The entire project will be compiled into a WAR file which will then be deployed on a host call Apache Tomcat server.

https://docs.spring.io/spring/docs/3.0.0.M3/reference/html/ch16s05.html