Bank of SUN Developers

Members:
Paul Theron
Justin Smith
Heinrich De Lange
Ryan Evans

Table of Contents

1 Version Control.... 4
1.1 BankOfSunBRD.... 4
1.1.1 Properties.... 4
2 Executive Summary/Introduction.... 5
3 Business Use Cases.... 6
3.1 Business Use Cases.... 6
4 Business Use Case Descriptions.... 7
4.1 BUC - Application.... 7
4.2 BUC - Notifications.... 8
4.3 BUC - User Functions.... 8
4.4 BUC - Query System.... 9
5 Actors.... 10
6 Role Map.... 10
6.1 Role Map.... 10
7 Systems Use Cases.... 11
7.1 SUC - User Functions.... 11
7.2 SUC - Application.... 12
7.3 SUC - Notifications.... 12
7.4 SUC - Query System.... 12
8 Systems Use Case Descriptions - Activity Diagrams, Wireframes and Textual Descriptions.... 13
8.1 SUC - Application: Select apply option 14
8.2 SUC - User Functions: Buy from external vendors.... 15
8.3 SUC - Notifications: Sends push notifications.... 16
8.4 SUC - User Functions: Open new account.... 16
8.5 SUC - User Functions: Notify Clients.... 17
8.6 SUC - Query System: Select FAQ.... 18
8.7 SUC - Query System: Select create ticket.... 19
8.8 SUC - Query System: Answer ticket and respond.... 20
8.9 SUC - User Functions: View Applications.... 20
8.10 SUC - User Functions: Make payments.... 21
8.11 SUC - User Functions: Update profile.... 21
8.12 SUC - User Functions: Add beneficiary/recipient.... 22
8.13 SUC - User Functions: View balances on accounts.... 22
8.14 SUC - User Functions: View client list.... 23
9 State Machine Diagrams.... 23
9.1 SMD - Query Ticket.... 23
9.2 SMD - User Application.... 24
9.3 SMD - New Account.... 25

9.4 SMD - External Vendors..... 25
9.5 SMD - Payments..... 26
10 Class Diagram..... 27
10.1 Class Diagram..... 28
11 Non-functional Requirements.....29
11.1 Performance Requirements 29
11.2 Stress Requirements 29
11.3 Response-time Requirements 29
11.4 Throughput Requirements 29
11.5 Usability Requirements 29
11.6 Security Requirements 30
11.7 Volume and Storage Requirements 30
11.8 Configuration Requirements 30
11.9 Reliability Requirements 30
11.10 Backup/Recovery Requirements 30
11.11 Training Requirements 31

1. Version Control

1.1. BankOfSunBRD

1.1.1. Properties

Author	team return passModule(goodMarks)
Company	
Description	<table><tr><th>Version</th><th>Date</th></tr><tr><td>1.0</td><td>5 October 2018</td></tr></table> **Participants:** Paul Theron Justin Smith Heinrich De Lange Ryan Evans
UML Version	2.x

2. Executive Summary/Introduction

Executive Summary:

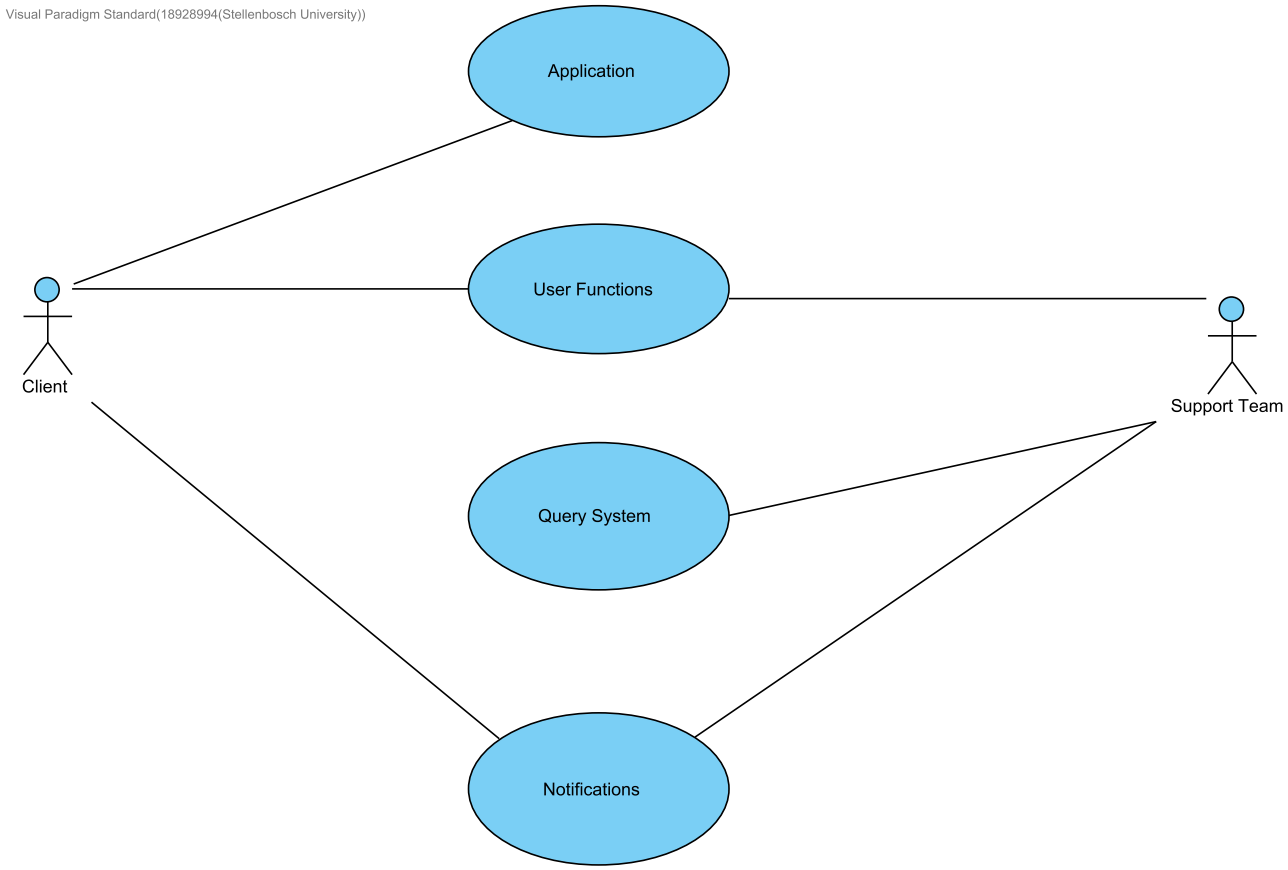
The following BRD was created to show the necessary working components that are required to build a functioning website for Bank Of Sun. The website will be a new banking platform for the Stellenbosch community. The BRD draws attention to the different components working together consisting of Business Use Cases, System's Use cases with HTML mockups and more that represent what the prototype will look like. The BRD provides a consistent tone and logic we will use to build the functioning website and still be attentive to our client and company's goals. The BUC's represent our main goals with the website application, notification, payment/transfer, query system. The functionality of these goals represent how well we have completed it.

Introduction:

Bank of Sun is a new and exciting bank created by the philanthropist and entrepreneur Justin Smith. His vision was to make an inclusive bank where benefits and rewards are enjoyed by everyone regardless of their background. The bank started off as a small idea in the Stellenbosch launch lab as a meeting between friends a decade ago. Forward 10 years and Justin is finally realizing his dream of providing a sleek and user friendly banking experience. Now to streamline the experience to an even higher degree Bank of Sun has requested a functional and streamlined online experience. Free of any delays and other shortcomings that plague the online banking systems. We have provided a BRD outlining all our major goals. The BRD is a comprehensive and detailed document that consist of multiple parts displaying what we aim to do with Bank of Sun Online and that is to make it user-friendly to everyone. People with little knowledge of Online Banking and Investments need to be able to understand and use Bank of Sun without any difficulties or delays.

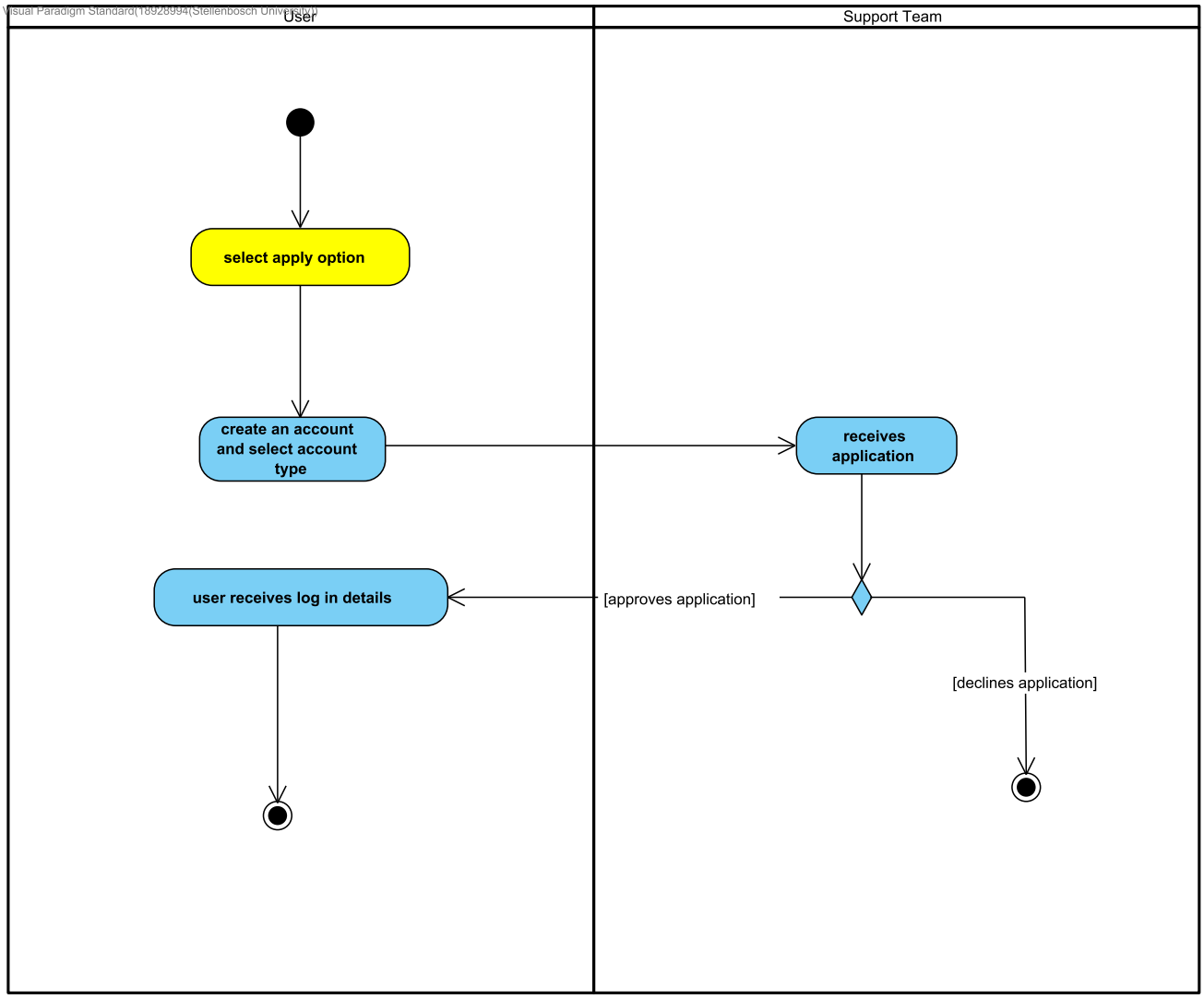
3. Business Use Cases

3.1. Business Use Cases

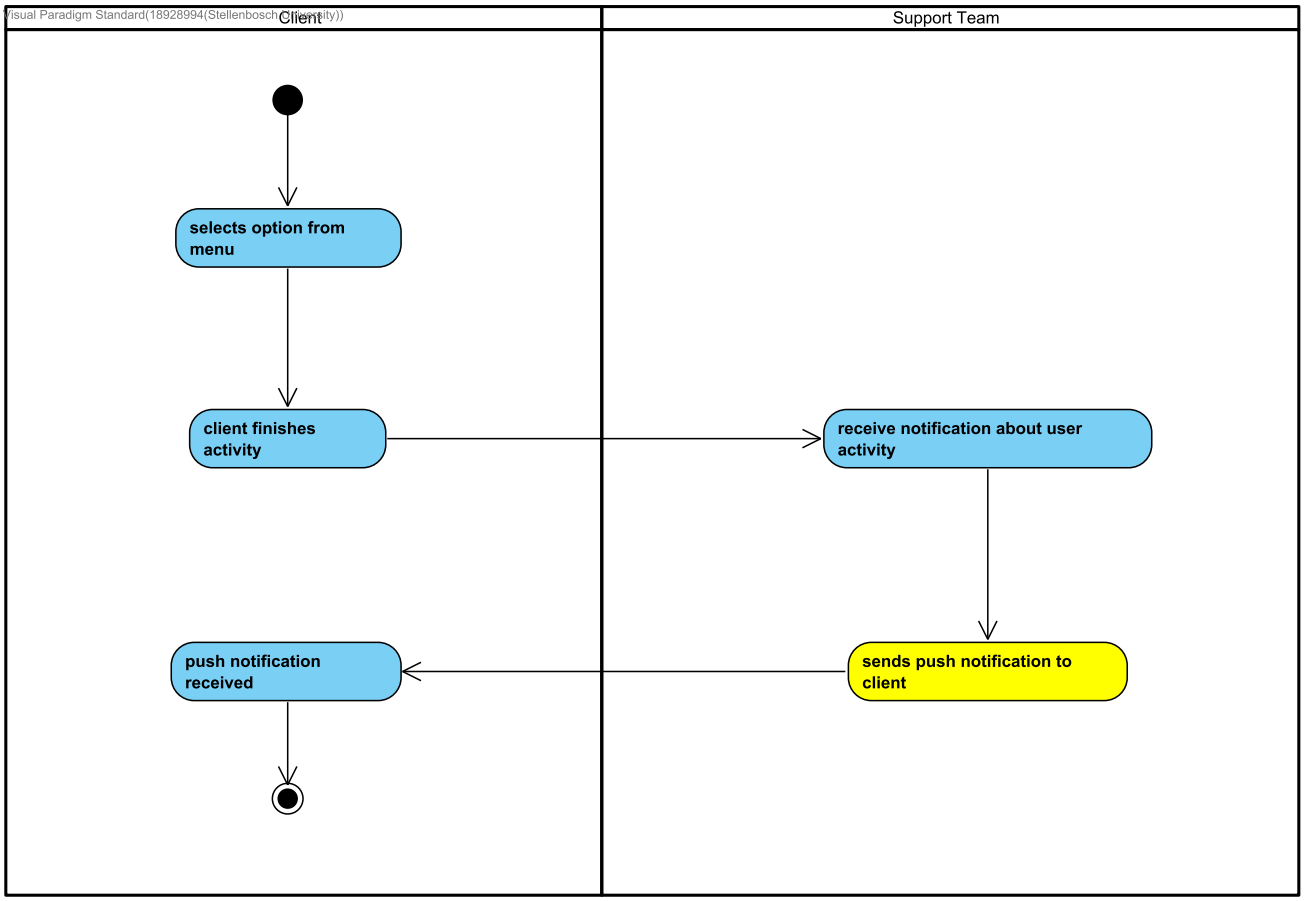


4. Business Use Case Descriptions

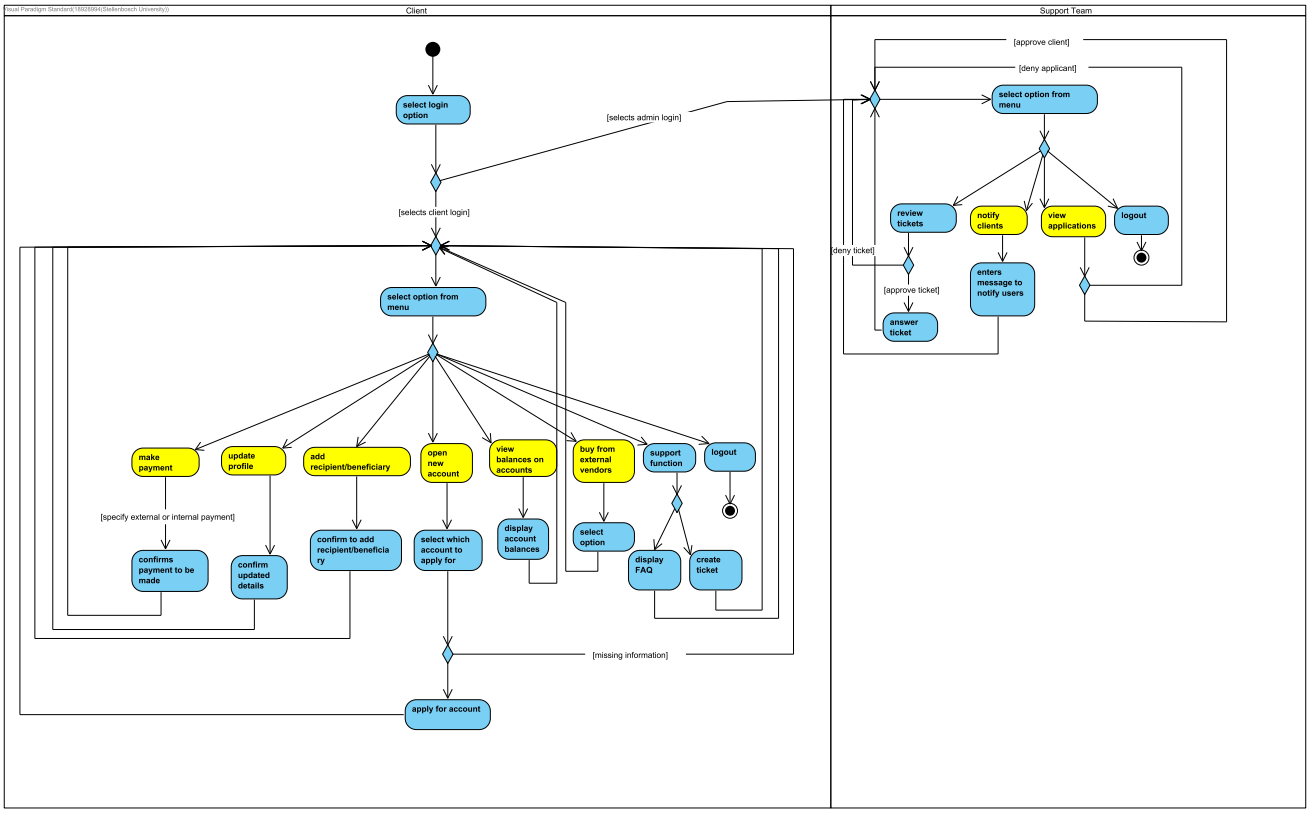
4.1. BUC - Application



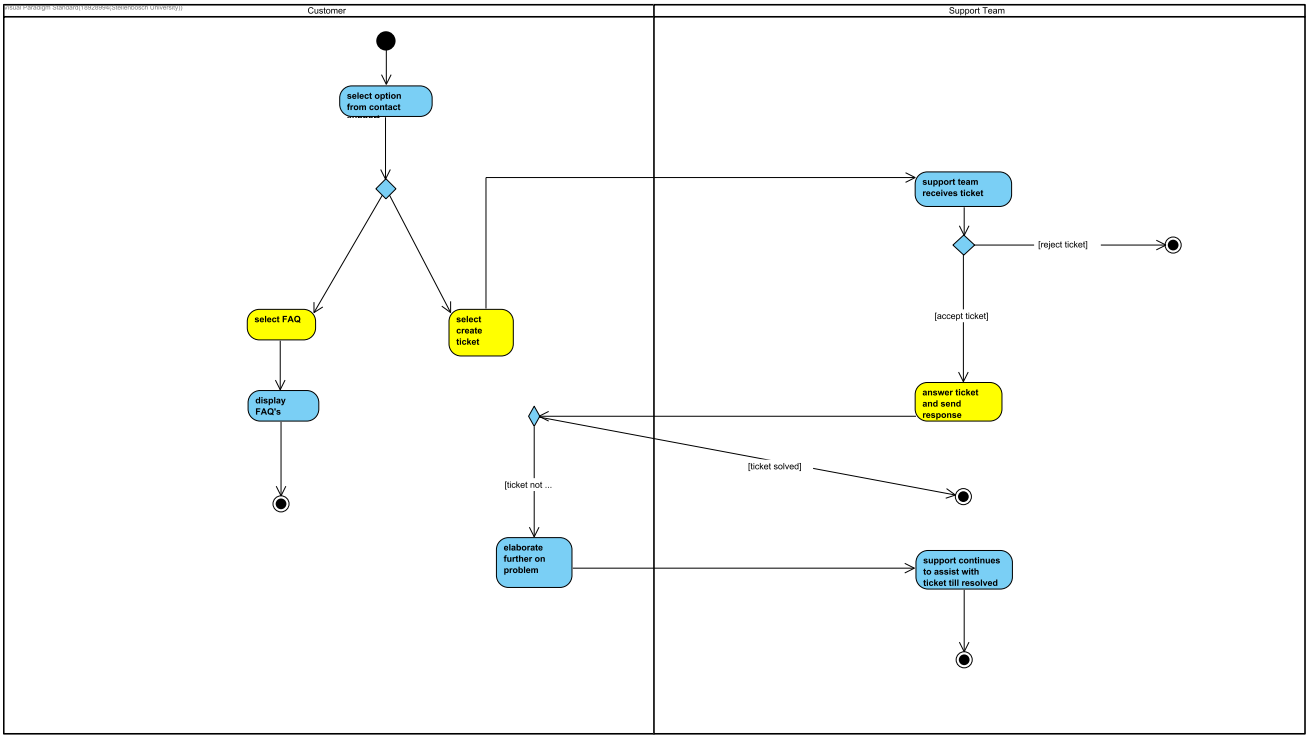
4.2. BUC - Notifications



4.3. BUC - User Functions



4.4. BUC - Query System



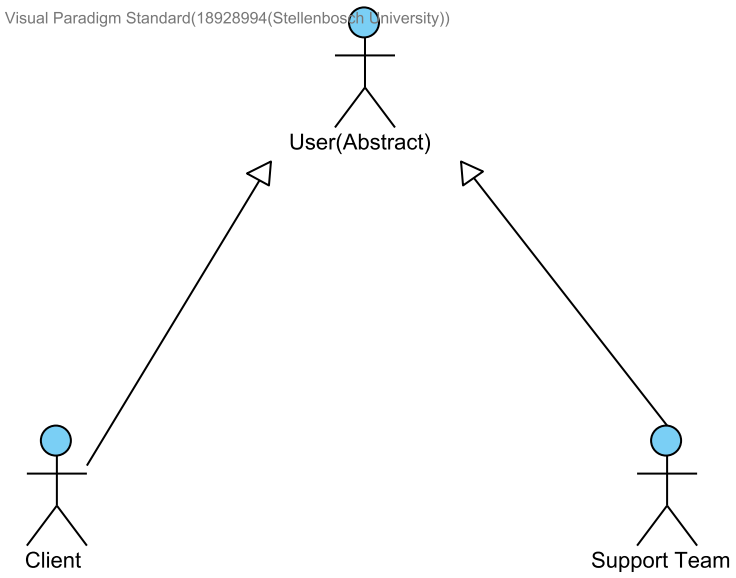
5. Actors

Here is a table containing all the different actors that are involved in the use of this application:

Position	Description
Client	The users that will be using the application on the client-side (perform different functions provided on the platform)
Support Team	The users that are internal actors which are compromised of Support Staff, Development team, tellers and the CEO. They will have access to the system and perform admin duties.

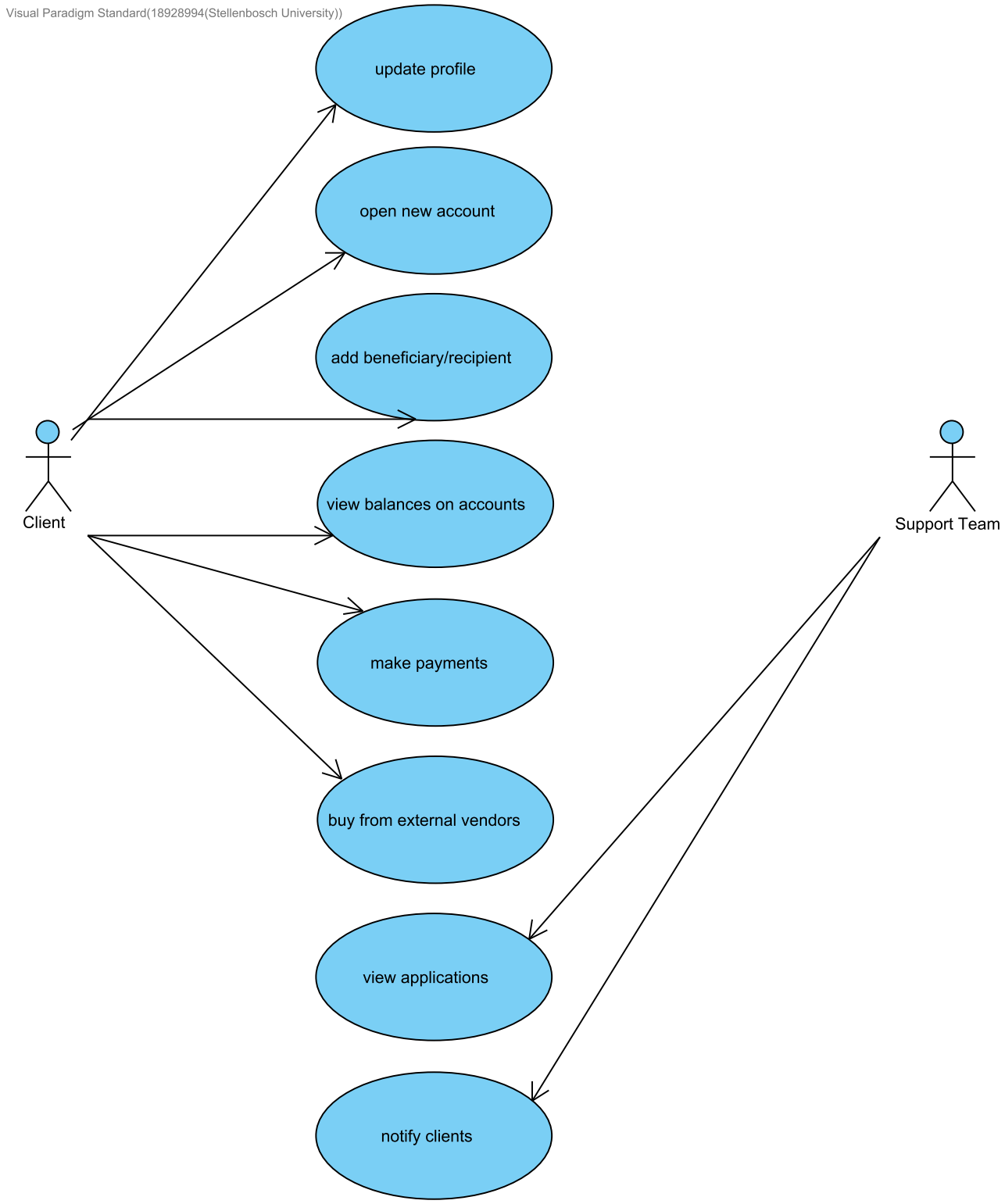
6. Role Map

6.1. Role Map

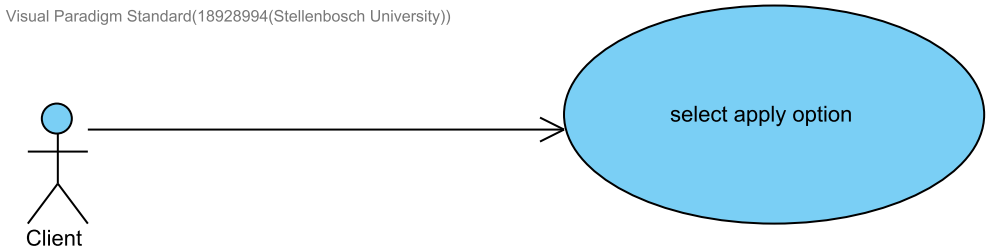


7. Systems Use Cases

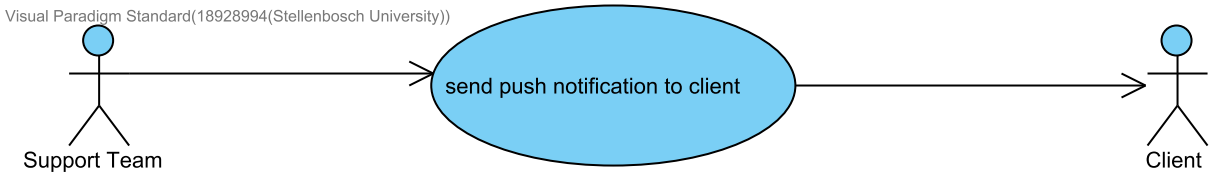
7.1. SUC - User Functions



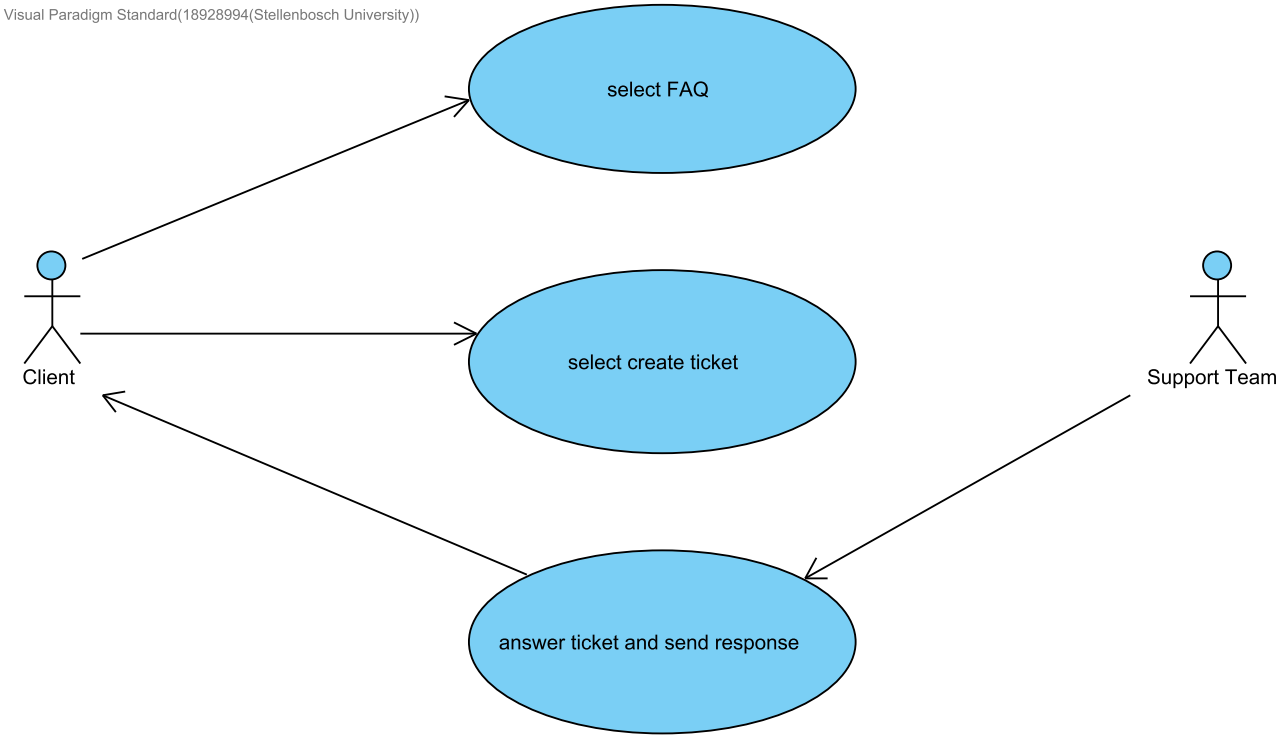
7.2. SUC - Application



7.3. SUC - Notifications

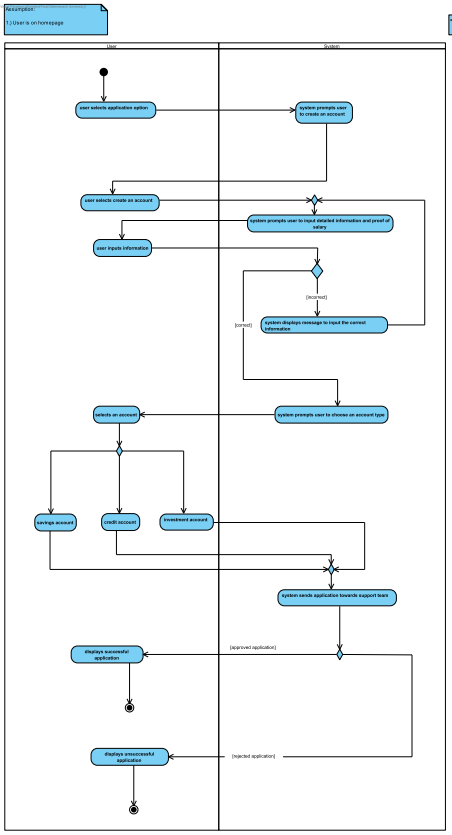


7.4. SUC - Query System



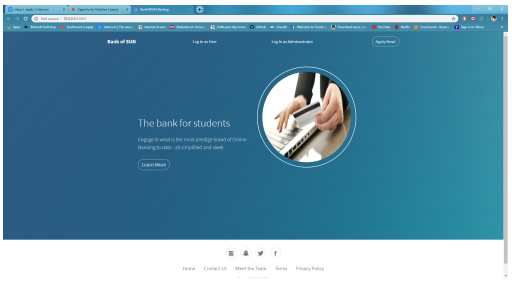
8. Systems Use Case Descriptions - Activity Diagrams, Wireframes and Textual Descriptions

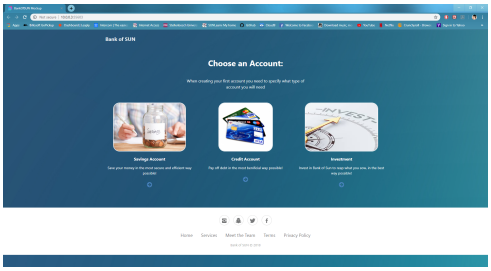
8.1. SUC - Application: Select apply option

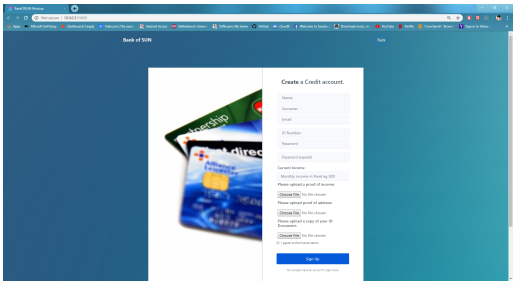


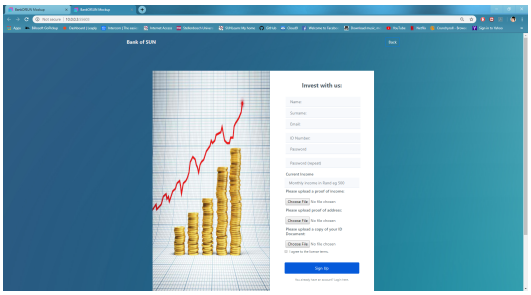
Textual Description

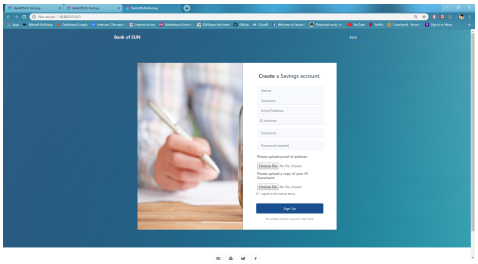
The user selects the application option from the homepage. The system then prompts the user to select an account type, and can choose from 3 different options. The user must specify which one. After specifying, the system will send the user application to the support team/admin where they can decide if the user will be approved or declined, upon receiving all the valid information.



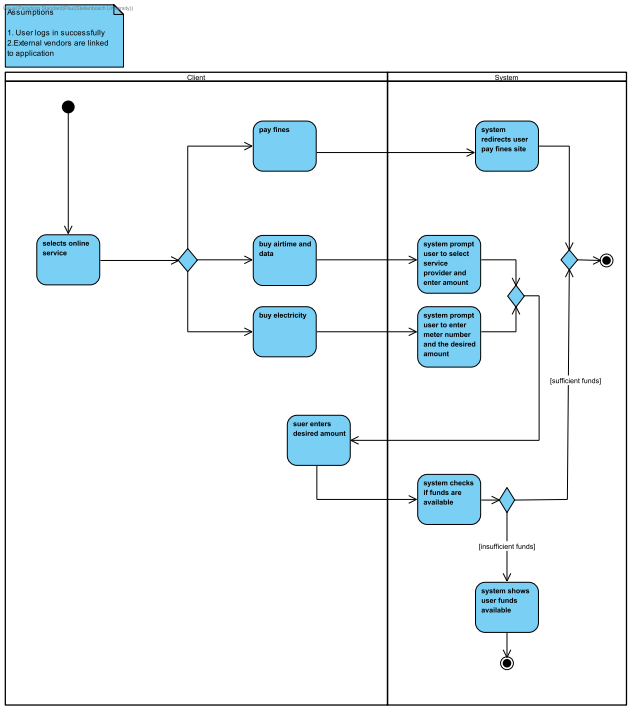






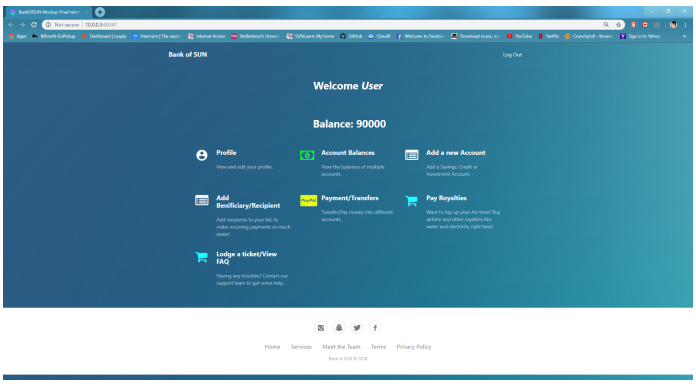


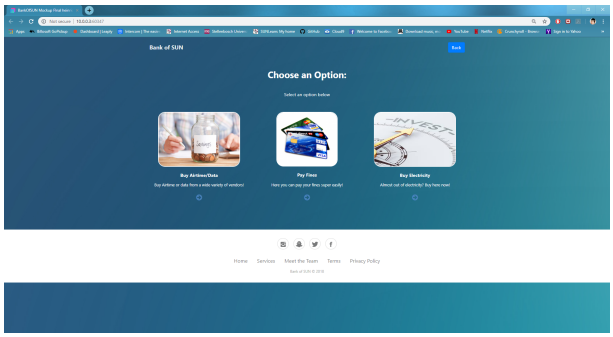
8.2. SUC - User Functions: Buy from external vendors



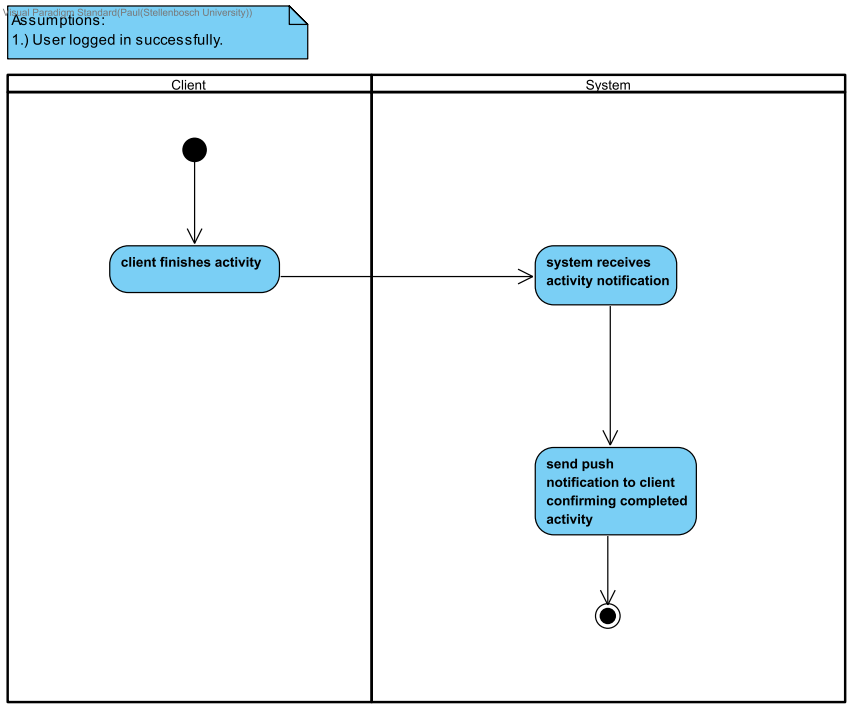
Textual Description

The user desires to buy from external vendors. User selects the buy from vendors option, and is prompted to select 3 different options - pay fines, buy airtime/data or buy electricity. When selecting to pay fines, the system redirects the user to go pay his fine on the traffic site. When buying airtime, the system prompts the user to specify his service provider and which amount of data/airtime he/she wishes to purchase. When the user selects the buy electricity option, the system asks the user to enter his meter number and the amount of units he wants to purchase. Each of these options goes through a error validation - to check if the account has sufficient funds to be able to complete these transactions. If enough funds are available, the action is completed - if not, it is declined.





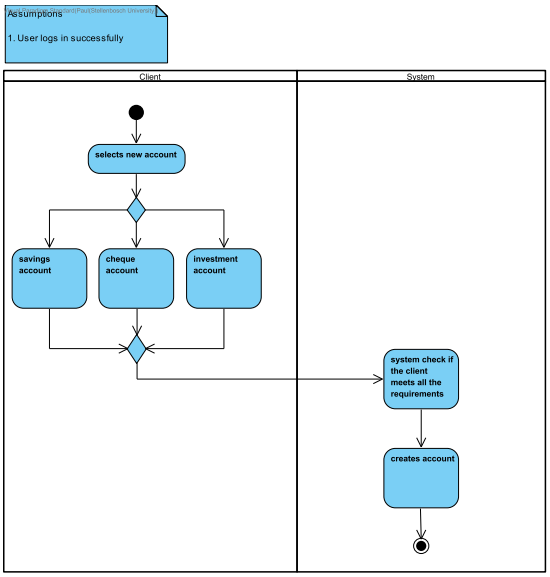
8.3. SUC - Notifications: Sends push notifications



Textual Description

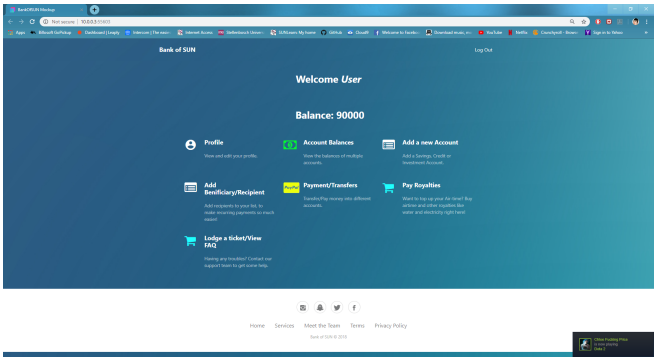
The user logs onto the system, and decides which option he wants to pursue.
After the user has finished the activity, the system is notified about the completion of the activity from the user side, and sends the user a push notification that the activity has been completed/declined.

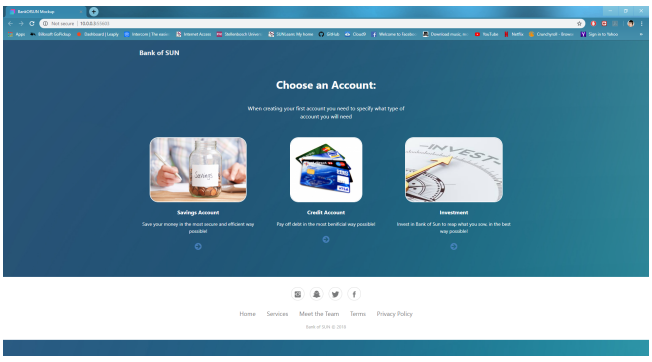
8.4. SUC - User Functions: Open new account



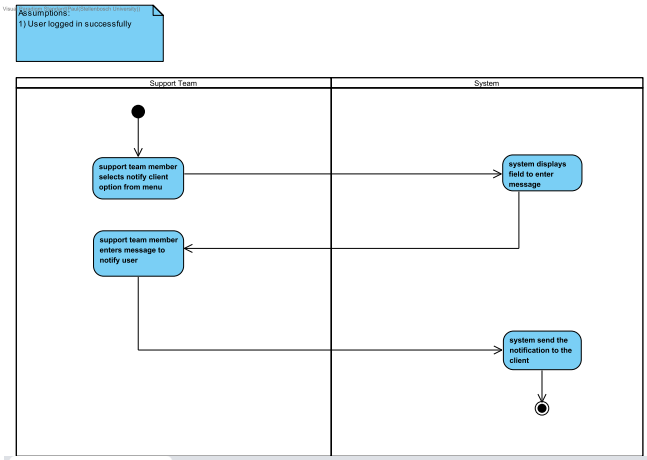
Textual Description

The user desires to open up a new account with the bank. He/she selects the apply for new account option, and specifies which account to apply for - Savings, cheque or Investment account. After the user specifies for which account he wants to apply for, the system checks if all the required information has been entered, and validates all documents. If everything is in order, the account is approved - if not, declined.



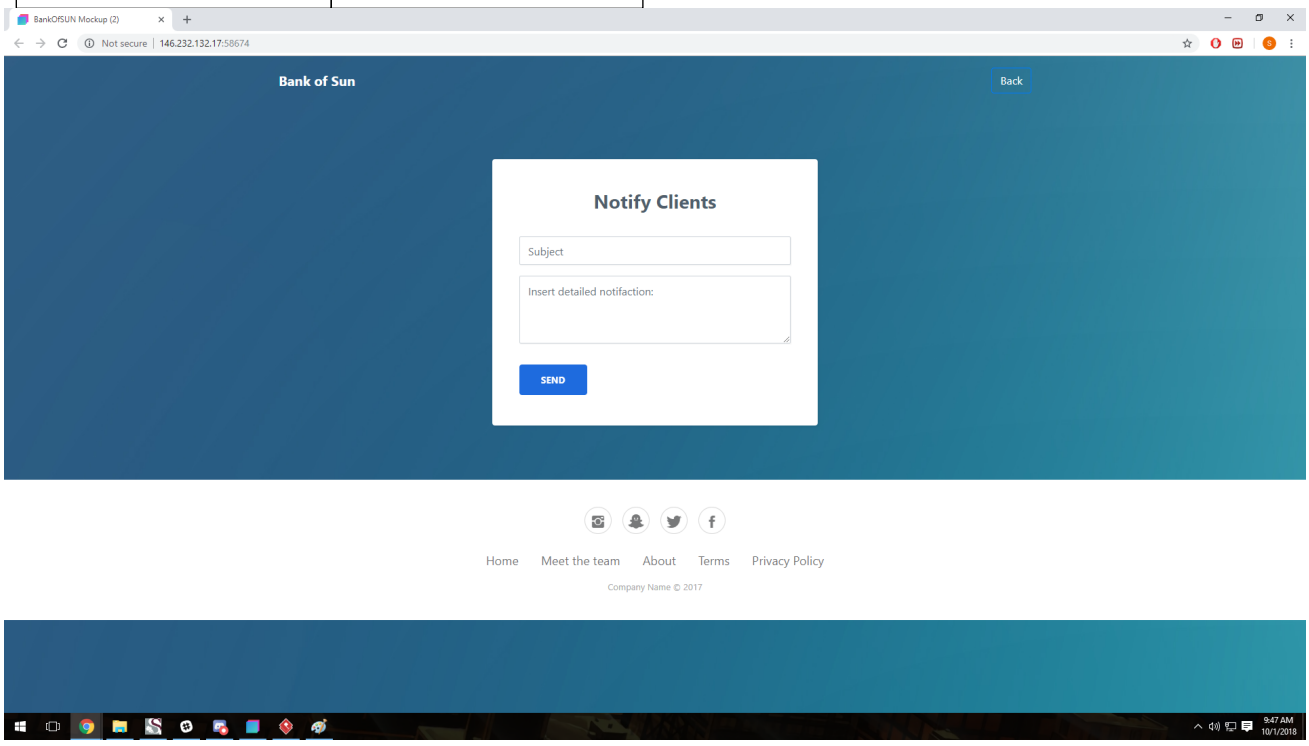


8.5. SUC - User Functions: Notify Clients



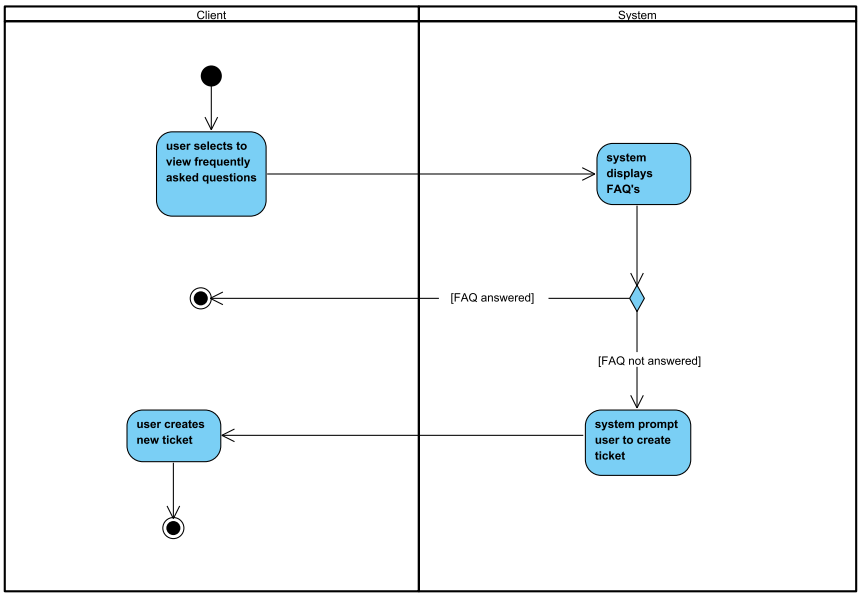
Textual Description

Support team member selects notify client from the menu. The system then displays a field for the user to enter a message into. The user then enters the message and the system then sends the notification to the client



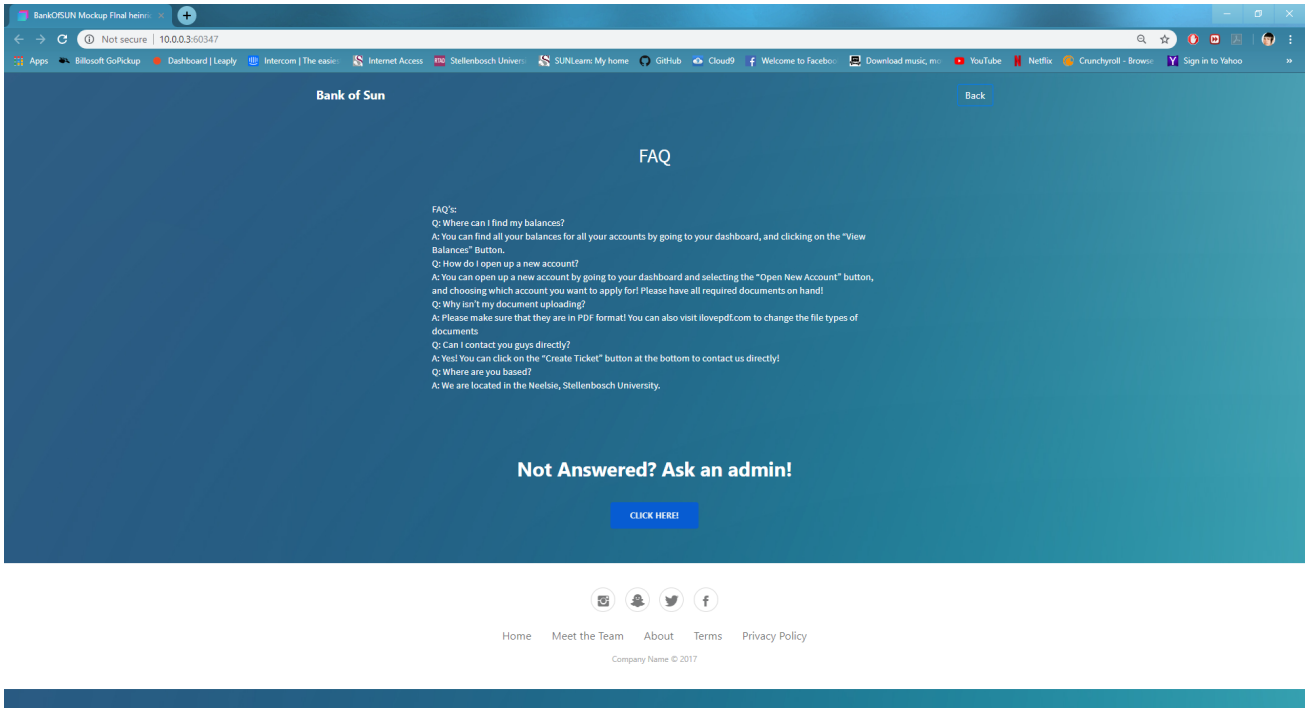
8.6. SUC - Query System: Select FAQ

Assumptions:
1) User is logged on successfully

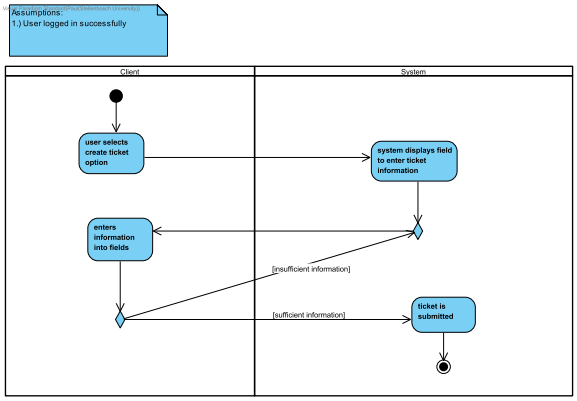


Textual Description

The user wants to view the Frequently asked questions (FAQ's) if the user is facing a problem. The user selects the support function option, and can then view the FAQ's. If the user is satisfied with the FAQ's, he can return to the main menu - if not satisfied, the system prompts the user to create a ticket if his/her answer is not solved.

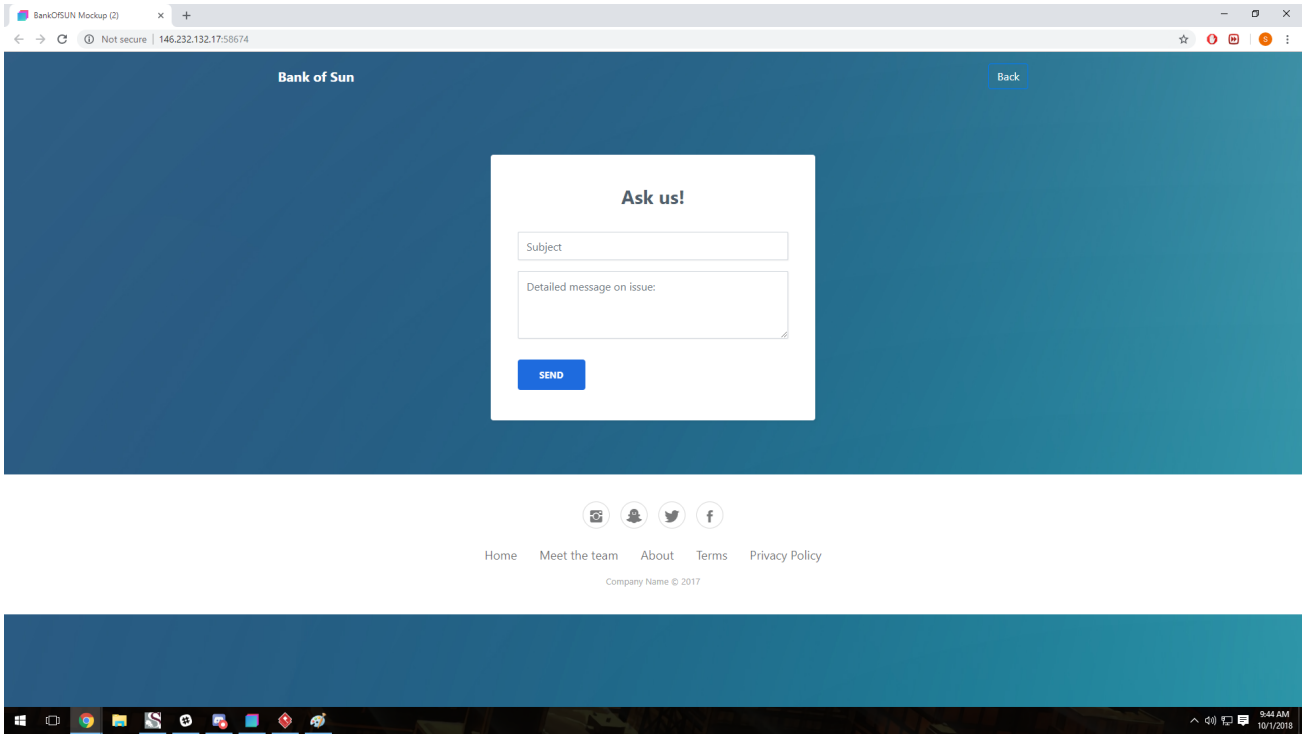


8.7. SUC - Query System: Select create ticket

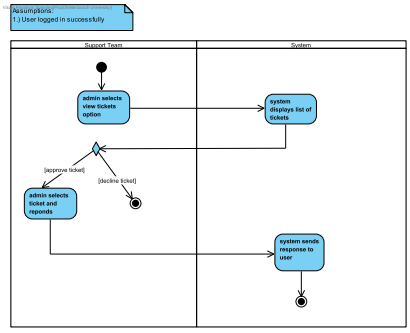


Textual Description

The user decides to create a ticket, after the FAQ's have not been helpful. Upon selecting to create a ticket, the system displays the fields where the user can enter his/her problem. After completing the fields, making sure that enough information is entered and no fields have been left blank, the system accepts the ticket and submits it to the admin team.

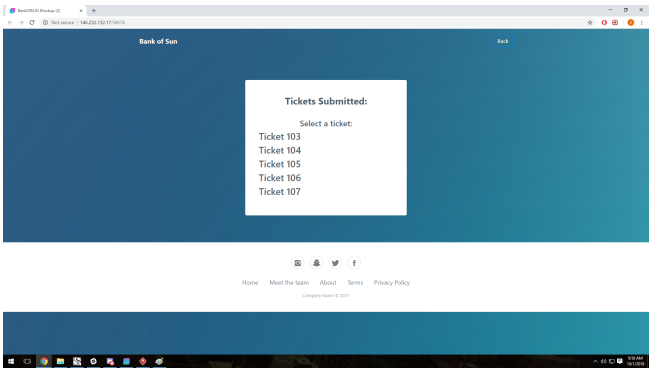


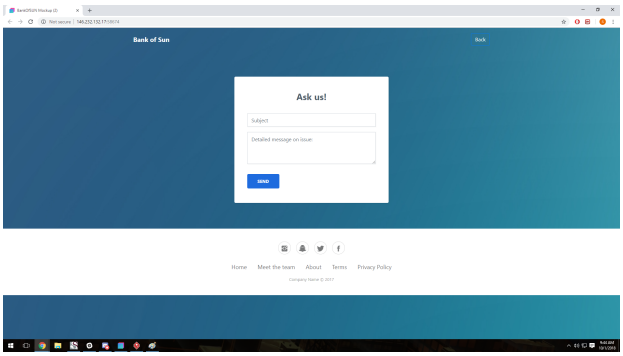
8.8. SUC - Query System: Answer ticket and respond



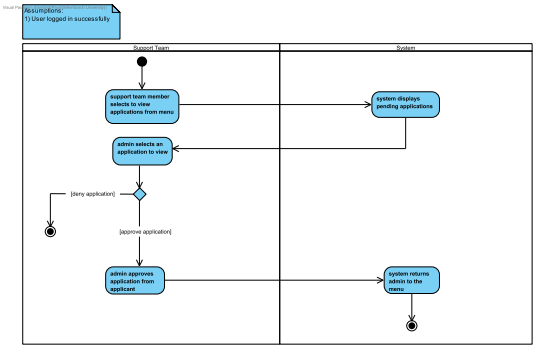
Textual Description

The support team is now logged into the system, and can select a variety of functions. The user decides to view the tickets. The system displays the list of tickets available and submitted. The user can then decide to approve and answer the ticket, or decline it. After the ticket has been approved, the user enters the appropriate answer and the system sends the response to the client.



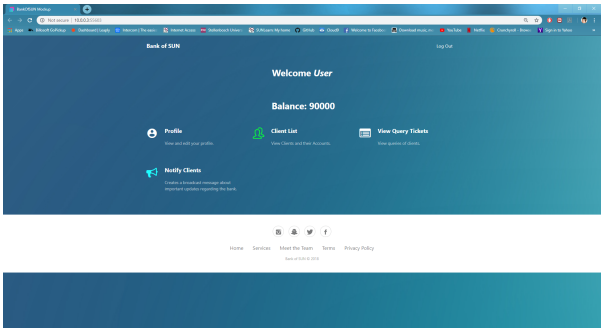


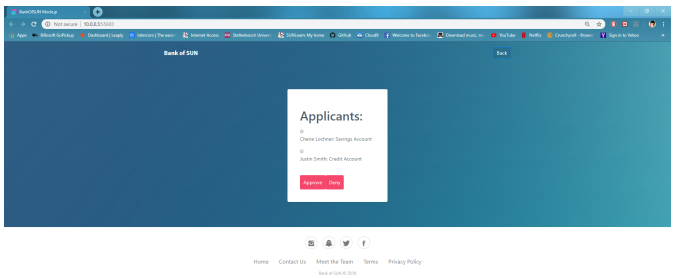
8.9. SUC - User Functions: View Applications



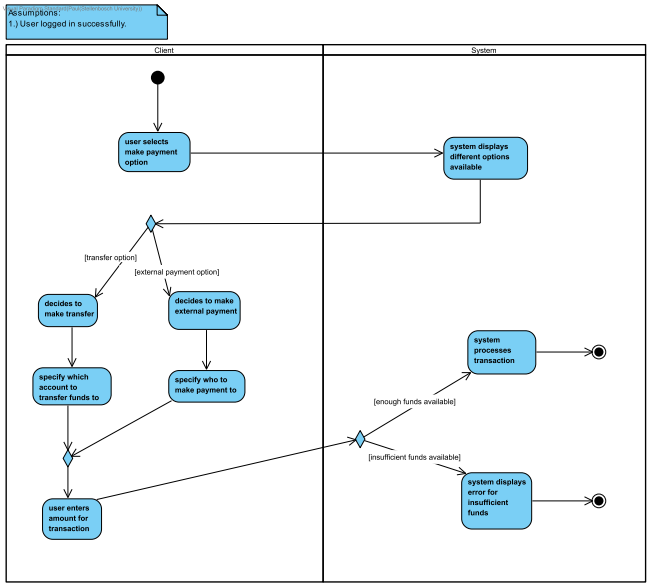
Textual Description

The user admin selects the option to view pending applications. The system then displays all pending applications. The admin is then able to deny the application or approve the application. If the application is approved a response is sent to the client notifying them of the result.



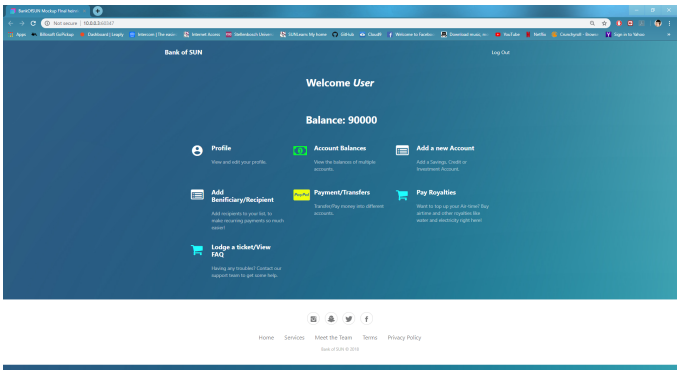


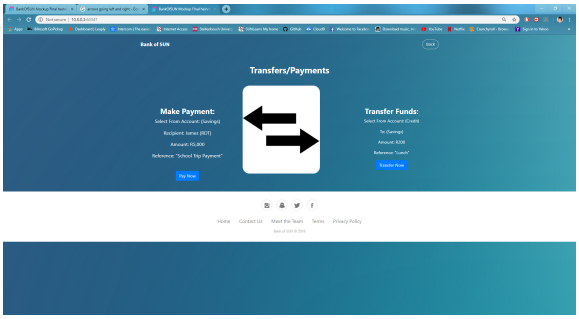
8.10. SUC - User Functions: Make payments



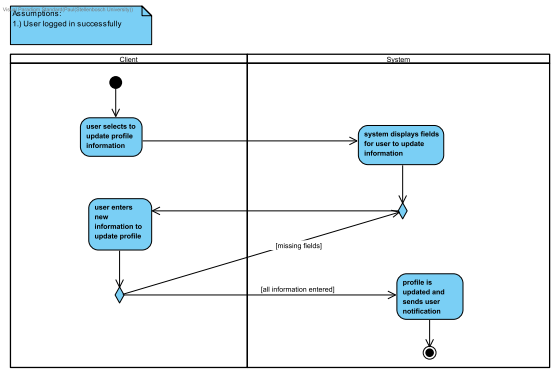
Textual Description

The user selects the payment option on the main menu. The system prompts the user to select from a variety of options - being transfer or external payments. If the user decides to make a transfer between accounts, the user is prompted to enter the amount to transfer and to specify which account, and the system calculates if enough funds are available and takes appropriate actions if not. If the user wants to make an external payment, the user needs to enter the amount to pay, and needs to specify who to make the payment to. After this, the system determines if enough funds are available and takes appropriate action if not. If enough funds are available, the system processes the transaction.



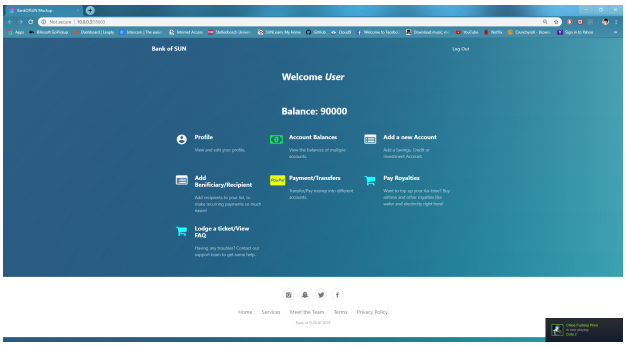


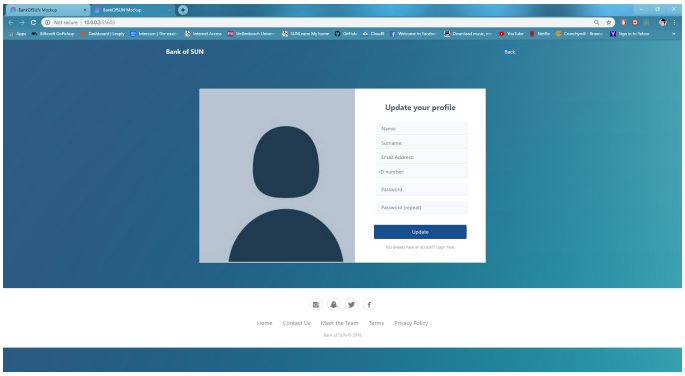
8.11. SUC - User Functions: Update profile



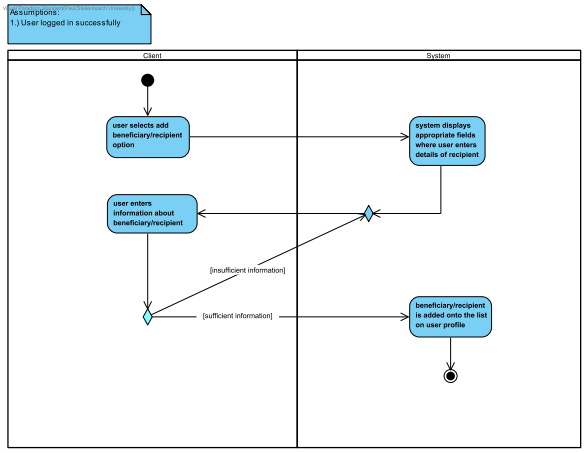
Textual Description

The user wishes to update his profile details. The user selects this option from the menu, and the system returns the fields where the user wishes to update his/her information. After the user has filled in the necessary information, the system validates that all info has been entered, and asks the user to confirm updated changes. When the user accepts, the system updates the users' profile and sends a notification that it has been updated.



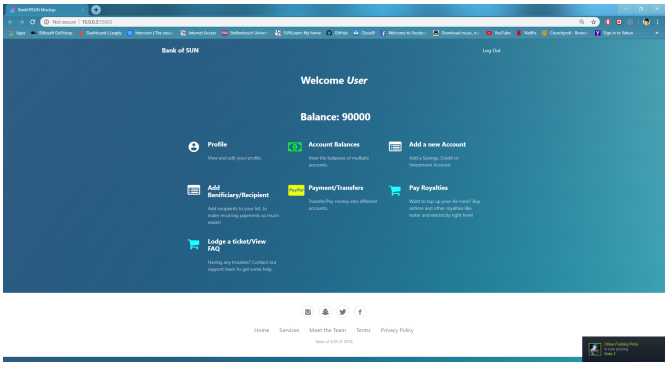


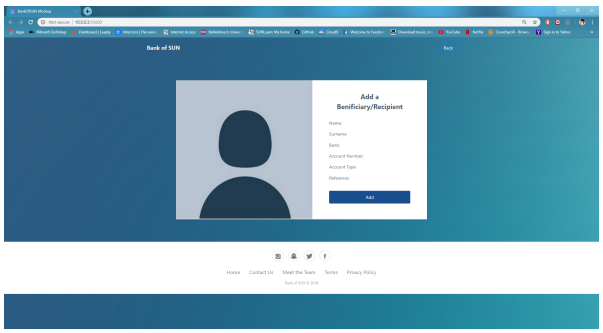
8.12. SUC - User Functions: Add beneficiary/recipient



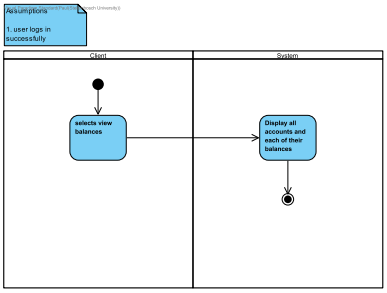
Textual Description

The user wants to add a beneficiary/recipient, so recurring payments can occur if needed. The user selects this option, and the system displays the appropriate fields where the user must enter all the information of the recipient, including details such as Full Name, Bank Type, Bank Account Number and Account Type etc. Once the user enters all the information, the system checks if all the fields have been filled, if not - the user is prompted to enter the information again, but if fully complete, the system adds the details of the beneficiary/recipient to the list so when a user wants to make a payment, they can select recipients from that list.



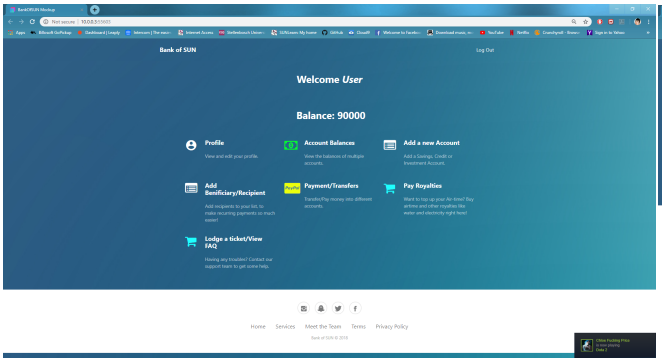


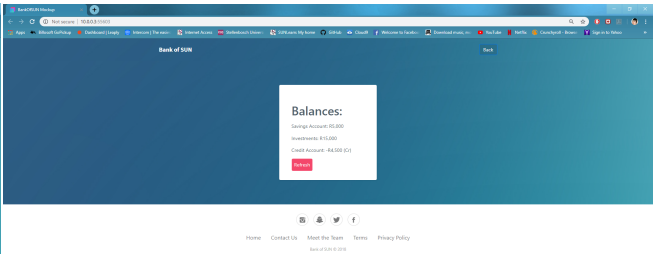
8.13. SUC - User Functions: View balances on accounts



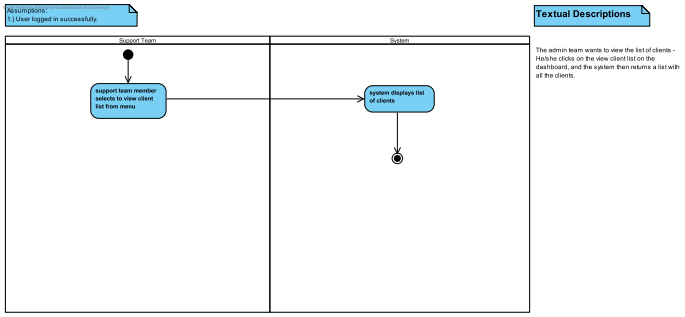
Textual Description

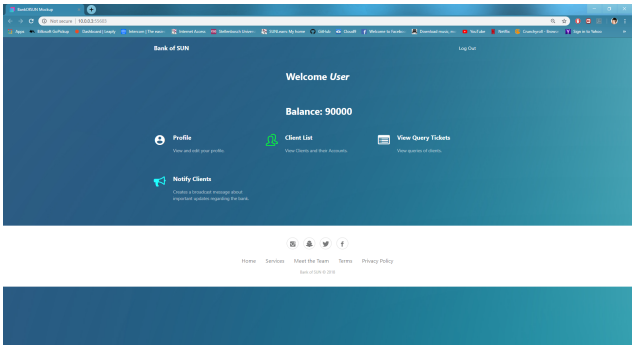
The user wants to view his/her balances on all accounts. He/she selects the view balances option - and the system displays all the balances for the account, including transactions.

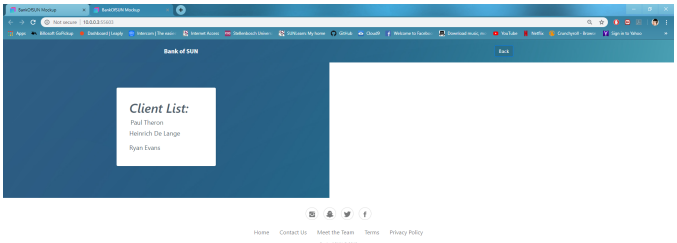




8.14. SUC - User Functions: View client list

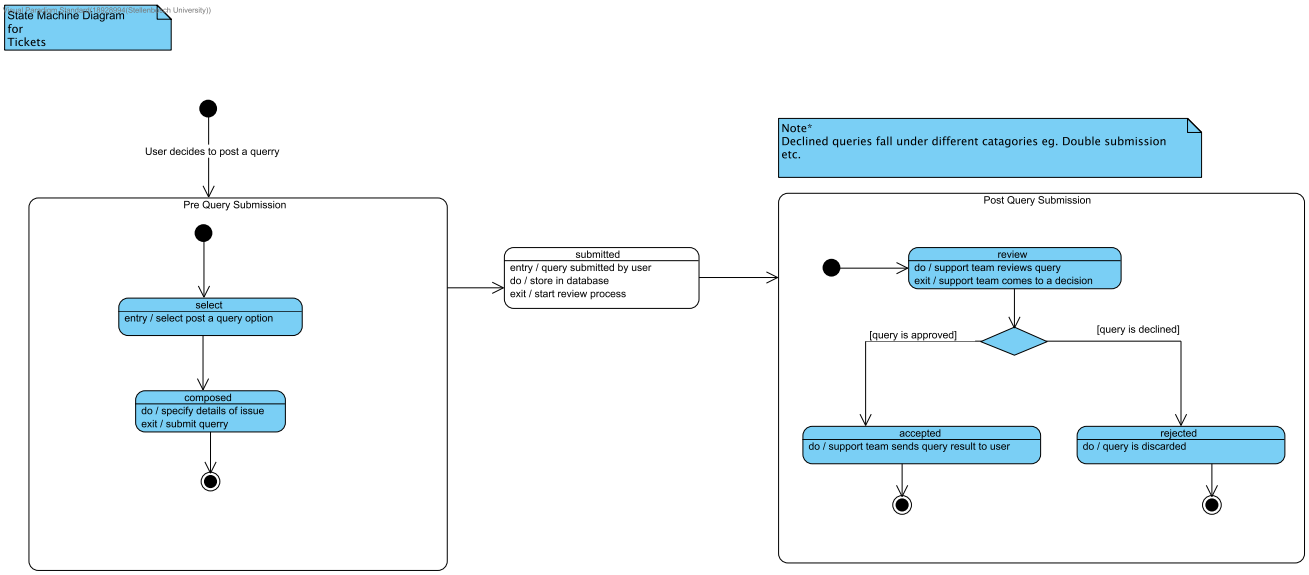




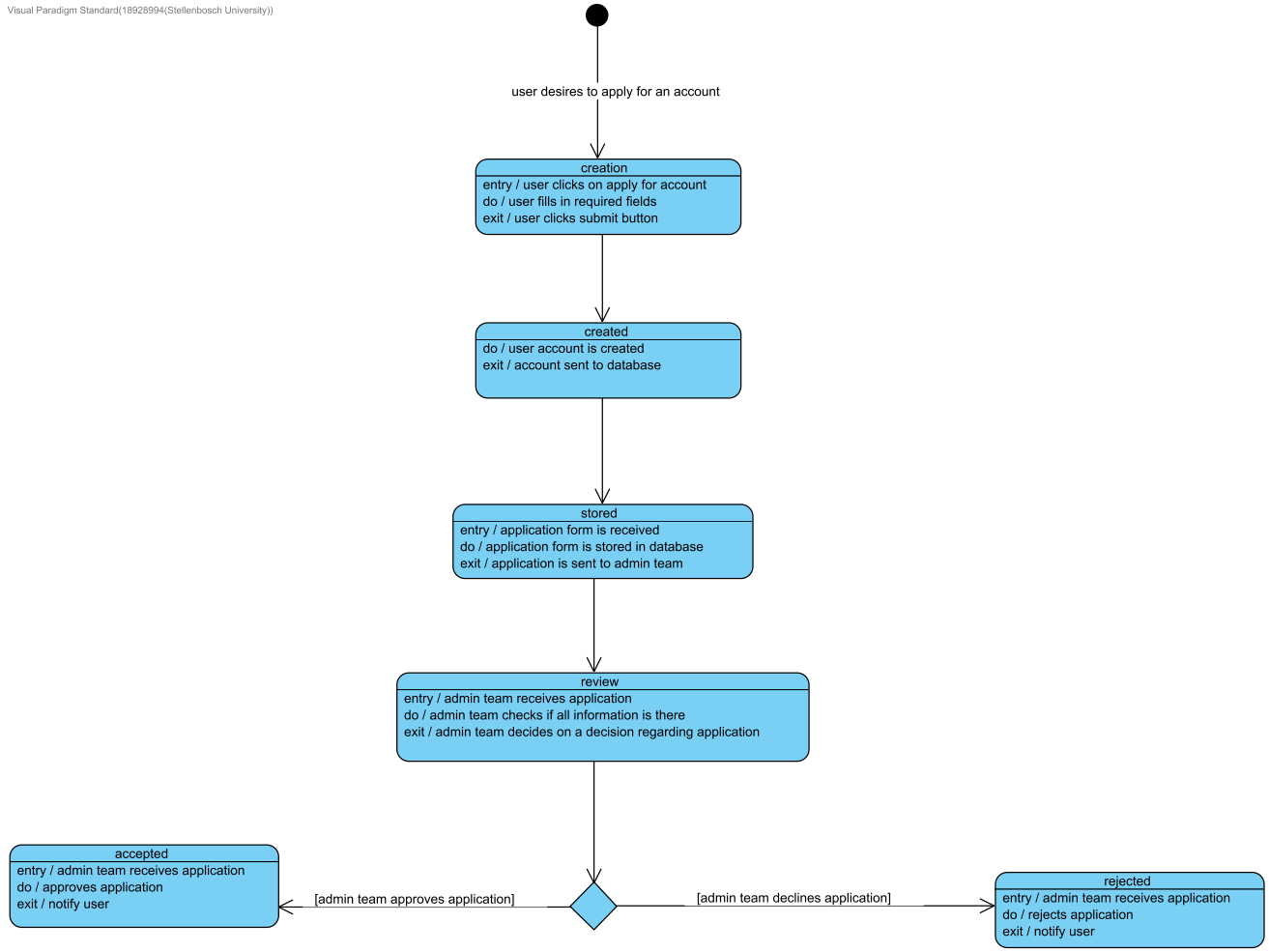


9. State Machine Diagrams

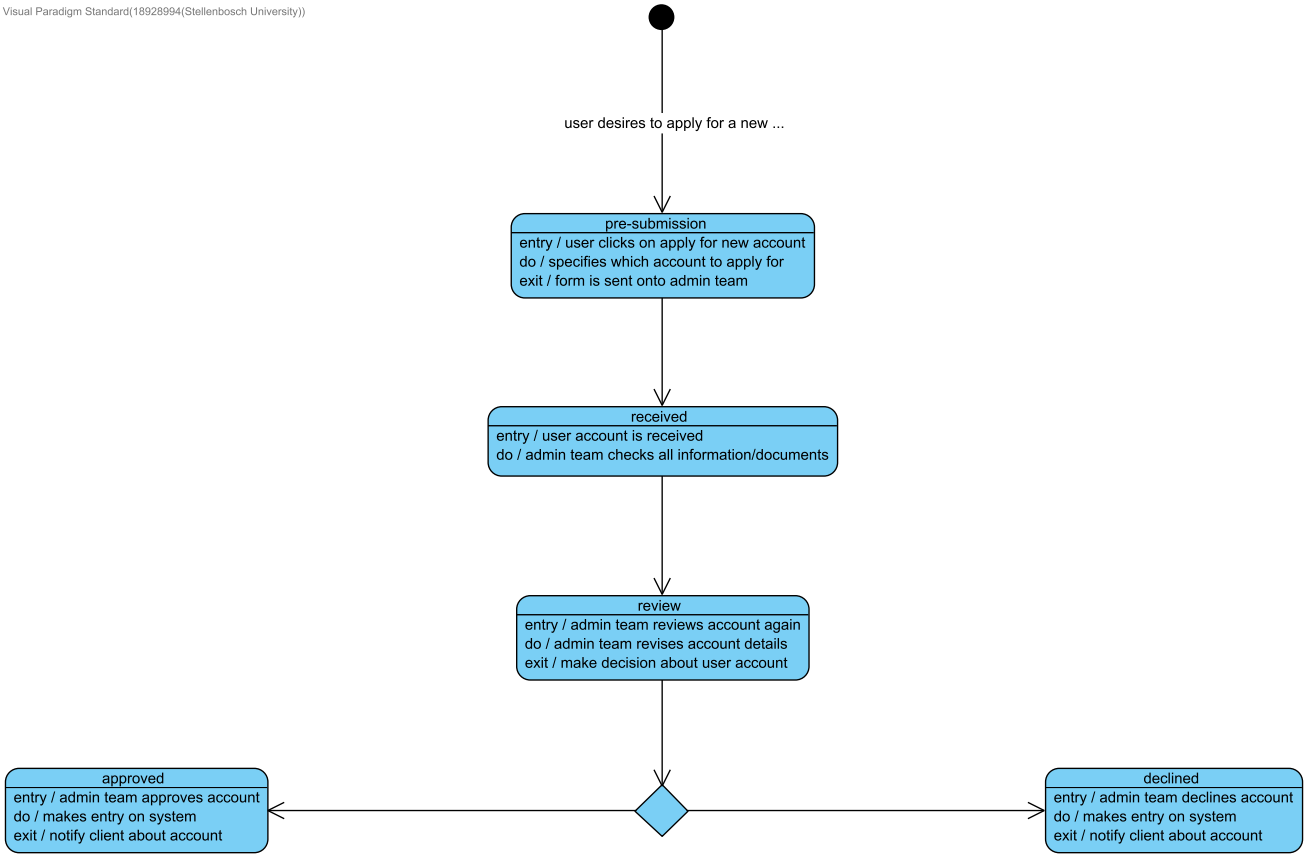
9.1. SMD - Query Ticket



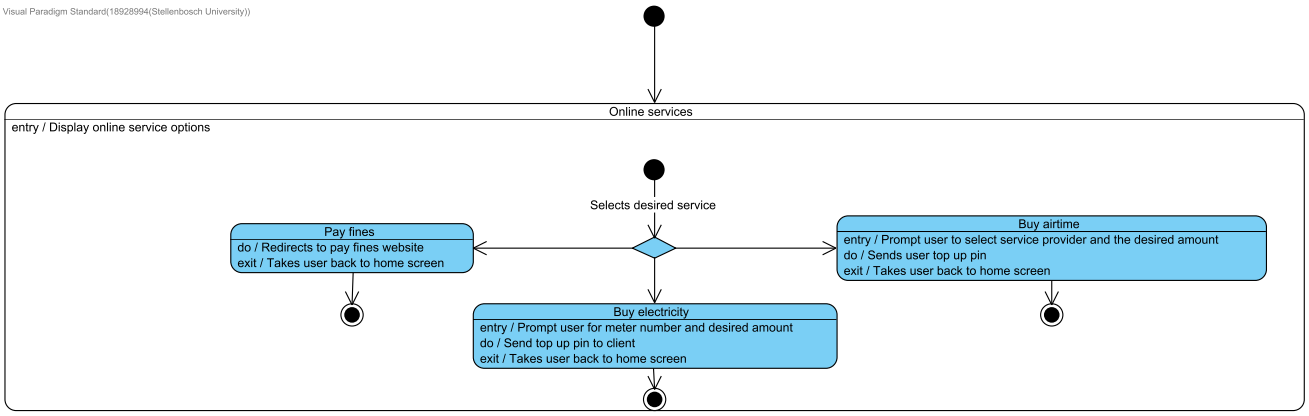
9.2. SMD - User Application



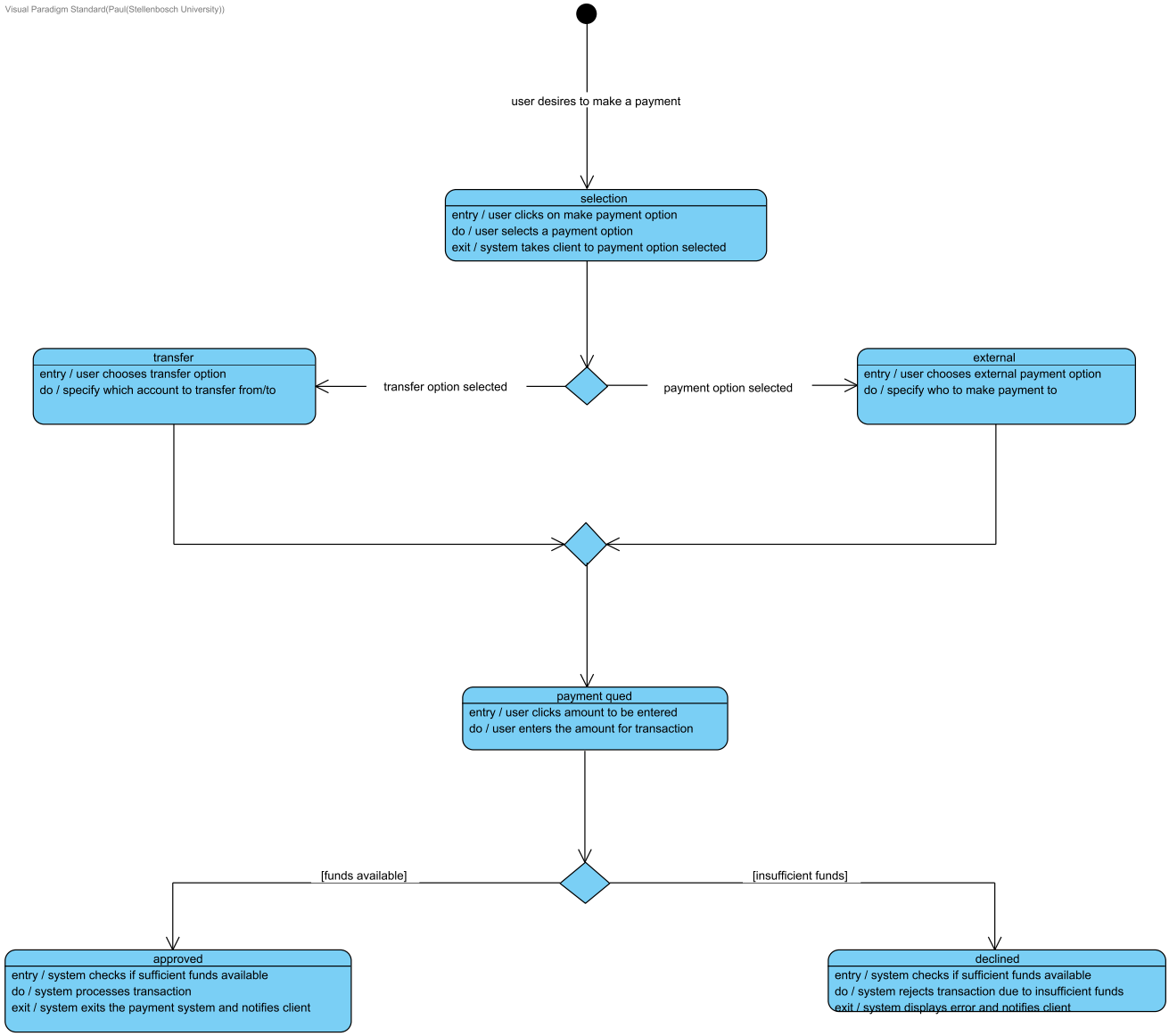
9.3. SMD - New Account



9.4. SMD - External Vendors

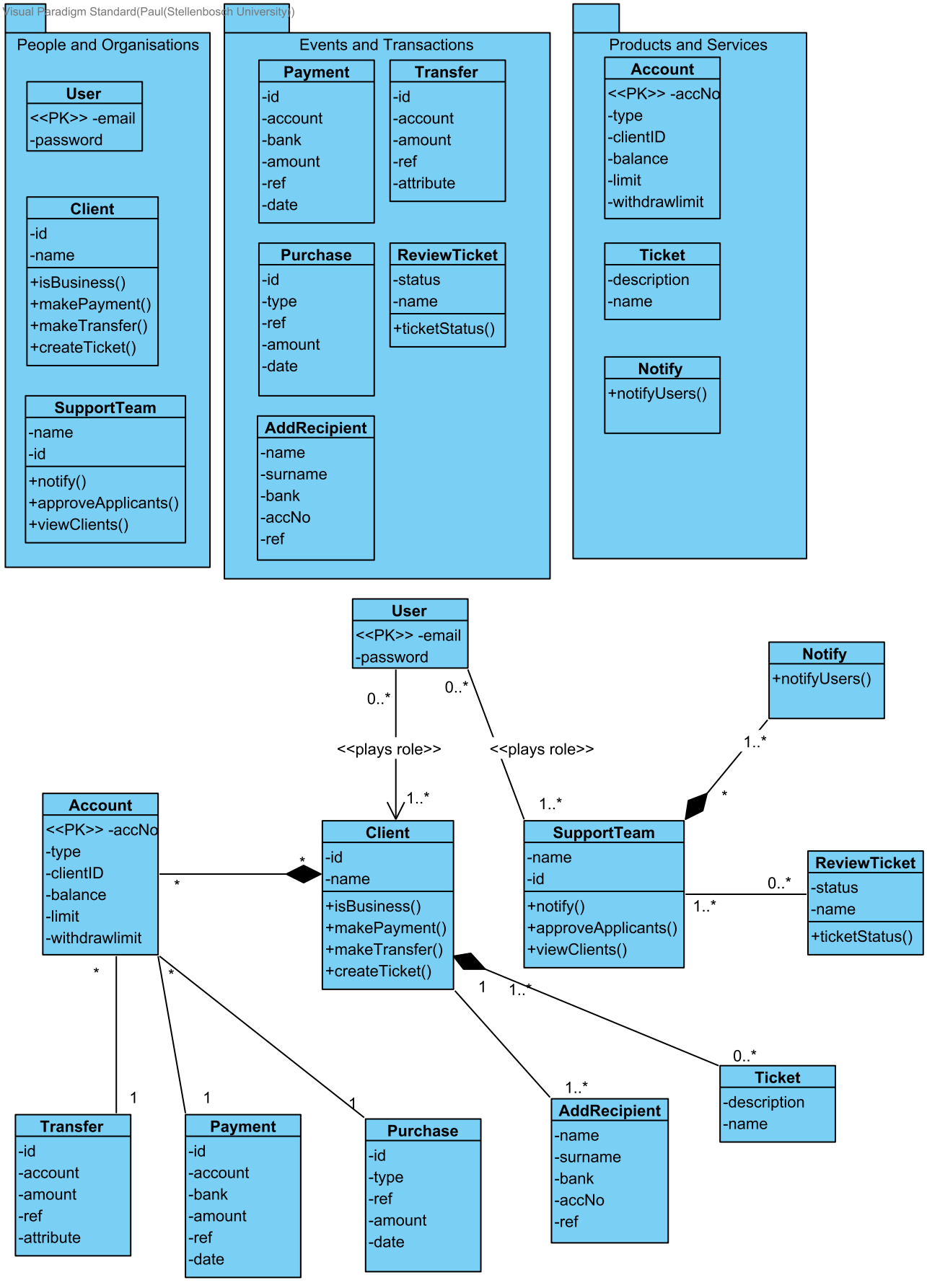


9.5. SMD - Payments



10. Class Diagram

10.1. Class Diagram



11. Non-functional Requirements

11.1. Performance Requirements

Mobile requirements:

Network: 802.11x (802.11b or higher).
Processor: 1.9GHz quad-core processor / 1.6GHz octa-core processor.
RAM: 1GB
Internal Storage: Website based application

PC requirements:

Network: 802.11x (802 11b or higher)
Processor: Dual Core @ 2GHz (i3,i5,i7) Or Any AMD Processor
RAM: 2GB Minimum
Internal Storage: Website based application

11.2. Stress Requirements

The application will run on a relative low amount of memory and processor speed. The application must be able to accommodate 1000 clients simultaneously.

11.3. Response-time Requirements

The response time from when a user submits a request on the application till the system delivers a supply is 30 minutes. (All requests must first be reviewed by the Support Team). The response time when using the app such as navigation is 2.0 seconds.

11.4. Throughput Requirements

The application shall be able to successfully process a minimum of 2000 user activities per day. The application shall be able to successfully handle 200 customer registrations per day.

11.5. Usability Requirements

A completely novice operator, given 30 minutes of training, must be able to complete the following functions without assistance: create an account, submit a ticket, apply for an account, make payments, transfer funds, add recipient and buy from external vendors.

11.6. Security Requirements

The application will conform of Smartphone Secure Development Guidelines by
ENISA. All security implementations are outlined in the following
document: https://www.enisa.europa.eu/publications/smartphone-secure-development-guidelines/at_download/fullReport.

11.7. Volume and Storage Requirements

The system must be able to store 5000 accounts. Accounts that are outdated or that is no longer being used must be removed automatically to free up space. The application will run on any device with more than 2GB of RAM.

11.8. Configuration Requirements

Mobile requirements:

Display type: Super AMOLED Plus capacitive touchscreen, 16M colors
Display Type: Touchscreen
Resolution: 480 x 800 pixels (~217 ppi pixel density) Multitouch: Yes.
Platform: iOS, Android, Linux and Windows

PC requirements:

Internet Connection required
Any applicable web-browser (Preferably Google Chrome)

11.9. Reliability Requirements

The application will run on 99 percent reliability. In each 24-hour timeframe, the application can fail only for 1 percent of the day while in use. 15 Minutes of the day an error may occur.

11.10. Backup/Recovery Requirements

All the data that is used and acquired from the application will be stored in a API. Only admin will have access to the database, the database will also be remotely stored on a server for improved security. In the event data failure or destruction, the developers can be contacted to gain permission to access the secured database for recovery purposes.

11.11. Training Requirements

No training whatsoever is needed to operate this application. The app will be developed in a manner that it simple, yet sophisticated and easy to use for all levels of expertise.