SOEN 6461 - Software Design Methodologies

Deliverable 1

iGo Ticket Vending Machine

Professor
Prof. Pankaj Kamthan

Team P
Authors
Aryan Saxena
Apoorva Sharma
Deepanshu Sehgal
Saumya Shah
Siva Nagi Reddy Segi Reddy

GitHub : `https://github.com/aryansaxena094/iGo.git`

Contents

Problem 1

1.1 Introduction

A ticket vending machine (TVM) is a self-service system which allows the users to purchase fare products for commuting from one place to another. For the current project we are targeting particularly the Subway TVMs in the city of Montreal.[1] Société de transport de Montréal (STM) is a public transit corporation which operates, maintains, and develops an efficient conveyance network for a fair price. The traditional system here has several TVMs located at each metro stations where you can get your tickets via the TVM. The user-interface is user friendly and provides several options for the tickets as one-time, daily, weekly etc. Payment options here are cash, debit, and credit cards.

Furthermore, we are offering a better approach which is the iGo application which can be downloaded from the respective app stores of Android and iOS devices. The iGo app would provide you to create an account which would provide several solutions over problems like -

1. **Long Wait time:** The traditional TVMs had a big disadvantage as in the peak hours, it would get crowded which would result in waiting for a long period of time.
2. **Complicated process:** It might sometimes get complicated for a certain group of people to operate a TVM i.e., a kid or a senior citizen.

1.2 Characteristics of iGO

The characteristics of the iGo application comprises of –

1. **Transportation usage:** The application would track an individual's usage and through which it suggests whether daily, weekly, one way, two way, or other would work for you based on your previous data, also it would have the data of the STM so it could provide you the frequency of the trains running and also notify you when the peak hours are, plus incase of any service issues, it would notify you.
2. **Notifications:** The application for the TVM would provide the user all the notifications like the payment went through or not, how much amount they were charged, when does the ticket expires (in some cases).
3. **User friendly interface:** The TVMs would have a better GUI which would have different languages and would also provide audio speech for blind or people with disabilities.

Extremely High Performance, Minimal Latencies, High Traffic Management, Very Available, Secure, and Accurate are all requirements for the software system. The purchase of tickets is not included in the scope of this project; hence the payment terms are not covered by the requirements. There is, however, a new Payment option that can be extended for realization. Support for different Languages is one of the Software's additional features.

1.3 Assumptions

1. The TVM is positioned in visible locations such as the center of metro stations or near entrances/exits.
2. The machine's user interface supports multiple languages, including English and French, to comply with Canadian laws.
3. The TVM has network security features that ensure secure transactions and prevent fraudulent use.
4. The TVM is constructed with durable materials to prevent theft and vandalism.
5. Smart Touchscreen Panel Human Machine Interfaces are available in metro stations and for scanning and validating electronic tickets and opus card validity.
6. Tactile marking indented keypads are provided for visually impaired individuals who require voice assistance.
7. The electronic tickets sold by the TVM are compatible with the transportation network's ticketing system.
8. The TVM accepts different payment methods, including cash, credit/debit cards, and mobile payment apps.
9. The TVM is regularly serviced and maintained to ensure proper operation.

Problem 2

2.1 Problem Domain Model

A problem domain model is defined as a visual representation of the fundamental ideas and connections found in a given field of study or expertise. It helps in understanding the scope of the problem and identifying pertinent entities, their characteristics, and the connections or relationships between them. It is very essential to help explain requirements and ensure that all stakeholders have a shared understanding of the problem domain model that is created during the initial phases of the software development project. For the representation of the problem domain model, different diagrams are used such as class diagrams, data-flow diagrams, or entity-relationship diagrams. For software engineers, it is a crucial tool, as it helps to ensure that the problem domain which is worked on is developed precisely according to the requirements and constraints of the problem.

Below we have defined the classes, attributes, and their relationships of the problem domain iGo TVM worked on:

- **igO TVM:** The main purpose of the iGo TVM is to provide the consumers a convenient way to recharge their opus cards and purchase subway tickets without waiting in long queues. In the iGo TVM class, language and address are considered as the two main characteristics. The address is the place where the ticket vending machine is placed and is considered as one of the important attributes of the class since it tells the different locations where the TVM is placed like inside the subway stations, bus stops and other public places in the city. The language attribute in the class includes two languages which is according to the location which we are considering i.e Montreal. So, it has two languages English and French.

  iGo TVM has the following relationships with other entities in the domain, as indicated below:

  - iGo TVM is used by one-to-many Passenger(s).
  - Zero to Many ticket(s) are dispensed by one-to-many TVM(s).
  - One iGo TVM has only one Payment Gateway.
  - The zero to many metro cards can be recharged by on-to-many TVM(s).
  - One-to-many Receipt(s) can be generated by one-to-many TVM(s).

- **Passenger:** The end user or beneficiary of the iGo TVM.

  Passenger has the following the relationships with other classes in the domain model as described below:

  - One-to-many passenger(s) can use one-to-many iGo TVM(s).
  - One-to-many ticket(s) can be purchased by one-to-many passengers.
  - One passenger can have zero or one card.

- **Subway Ticket:**Hardcopy tickets or paper-based tickets can be printed by the iGo TVM. The tickets are dispensed out of the TVM with a limited time frame availability.

  Subway Ticket has the following the relationships with other classes in the domain model:

  - Zero-to-many ticket(s) can be dispensed by one-to-many TVM(s).
  - One-to-many passenger(s) can purchase one-to-many ticket(s).

- **Opus Card:** An Opus card is a smart rechargeable card that allows people to travel within Montreal using public transport services like Subway, buses and trains[wiki reference]. It can be easily recharged at any iGo TVM.

  Passenger has the following the relationships with other classes in the domain model as described below:

  - One user can have zero or one metro card.
  - Any number of opus card can be recharged by iGo TVM.

- **Payment Gateway:** Every iGo ticket vending machine (TVM) is linked to a specific payment gateway, which controls all daily payments made by passengers, whether they are purchasing tickets or topping off their metro cards. Customers have the option of paying with cash, coins, a debit card, or a credit card.

  Payment Gateway has the following the relationships with other classes in the domain model as described below:

  - There is a one-to-one relationship between iGo TVM and payment gateway i.e one iGo ticket vending machine has one payment gateway.
  - The payment gateway is linked to all the national banks for verifying and processing the payment.
  - One payment: gateway is used by one-to-many passengers.

- **Bank:** A bank is an organization responsible for regulating the banking sector, ensuring the legitimacy and validation of transactions made through credit/debit cards when purchasing tickets or recharging metro cards.

  Bank has the following the relationships with other classes in the domain model as described below:

  - One Payment Gateway of iGo TVM is connected to one-to-many Bank(s).
  - There is a one-to-one relationship between Bank and payment since multiple banks cannot authenticate the same payment.

- **Payment:** A payment refers to the transfer of money or value from one party to another, usually in exchange for goods or services. Payments can be made in various forms, such as cash, credit/debit cards, bank transfers, checks, and digital currencies. The process of making a payment typically involves authorization, authentication, and verification to ensure the security and accuracy of the

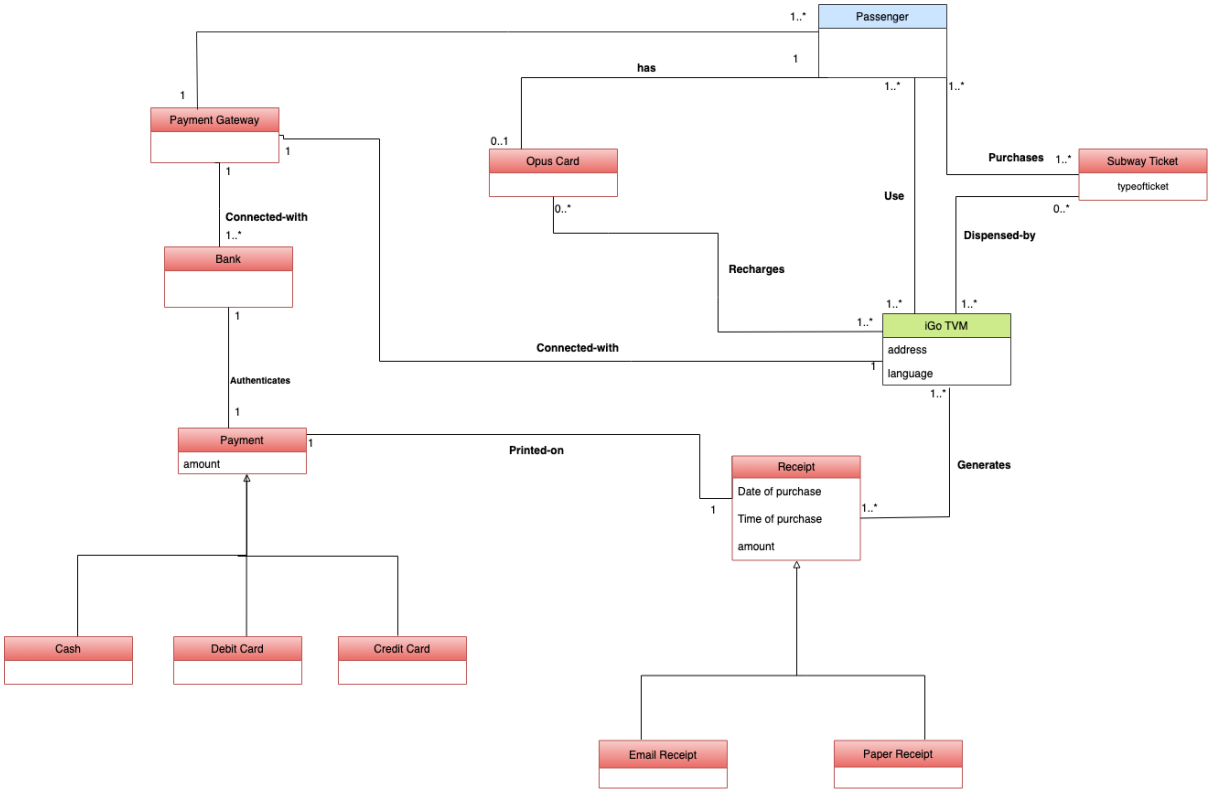

Figure 2.1: Domain model

transaction. Here in our problem domain the passenger can pay using cash, debit card or credit card.

Payment has the following the relationships with other classes in the domain model as described below:

- There is a one-to-one relationship as for a single session established through the payment gateway, one bank will authenticate one payment.
- Similarly, there is one-to-one relationship between payment and receipt as there will be one receipt for each payment.

- **Receipt:** A receipt serves as a proof of payment or transaction which contains details such as date, time and amount of the purchase.

  Receipt has the following the relationships with other classes in the domain model as described below:

  - One receipt only contains details of one payment.
  - One-to-many receipts can be generated by one-to-many TVM(s).

Problem 3

3.1 Problem: Mind map

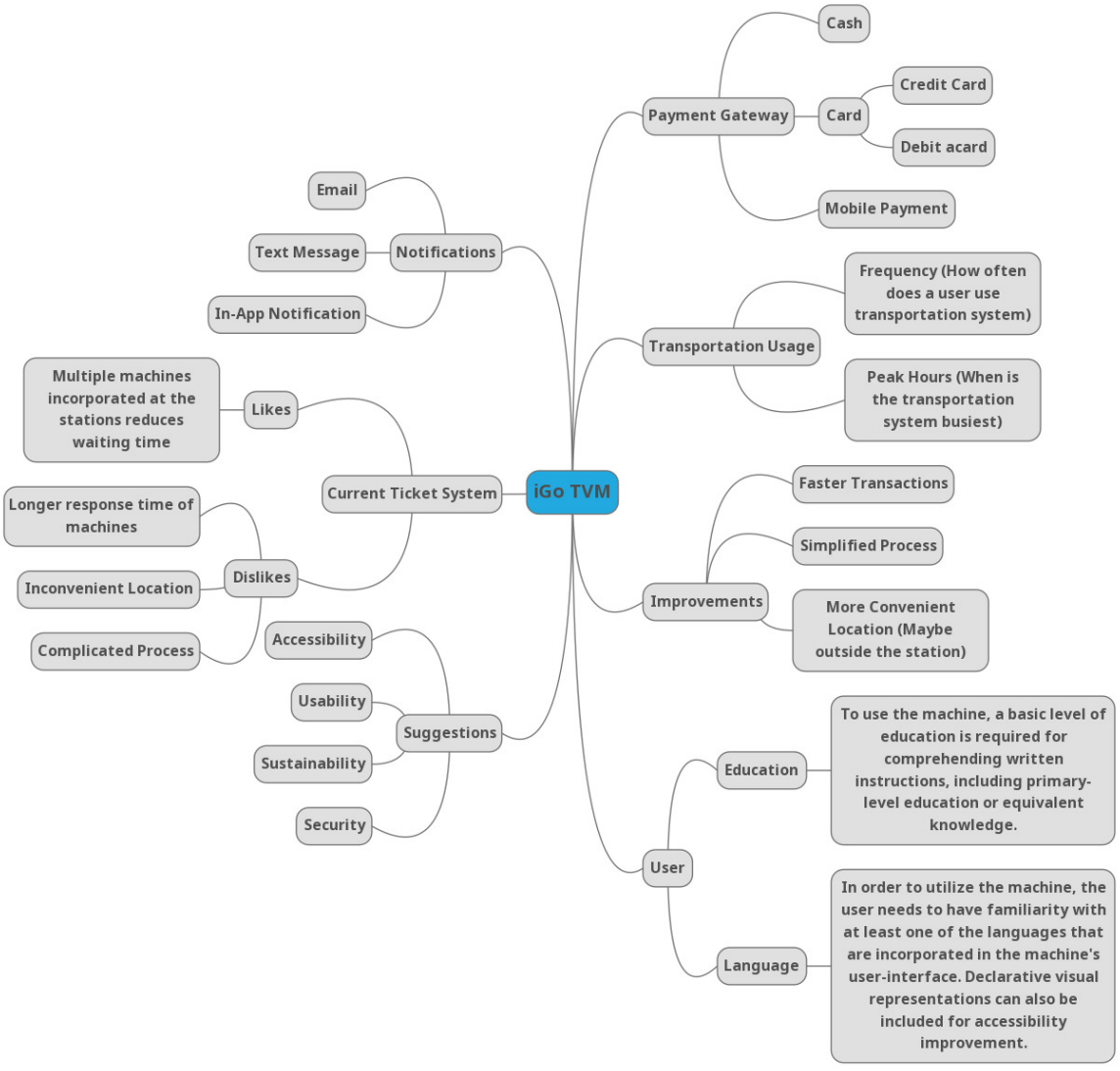

Figure 3.1: Mind map

Salient Conclusion of the interviews: We had fixed set of questions among the group members in order to establish a sense of uniformity among the questions that were asked from the respondents.

We tried to cover all the basics of the current TVMs and took insights as to what can be added in the iGo moving ahead. We also tried to keep the questions brief, crisp and short, as it would be difficult for respondents to make time from their life/busy schedules.

We inferred a lot of information from the interviews that were conducted by all the members of the group. Some concerns user specific and some were consistent. We have tried to jot down the ones we feel were consistent throughout the interviews conducted. Based on the responses we got for the questions we asked our interviewees we concluded that:

- **Firstly**, the users feel that navigating the interface in an unknown language is a hassle when it comes to operating a TVM. They want the first prompt to be the languages available so that they can choose and decide onto the language they want to use. As the respondents that were interview belong to Montreal, and none of them spoke French, they felt it was harder to navigate being unknown to the language and figuring out how to change the language.
- **Secondly**, most of the users felt that the TVMs should be placed near the ticket window or near some operator so that it would be easy for the person to ask for assistance. When an error occurs, either during transaction or getting the ticket, users feel insecure leaving their card inside the machine itself. Additionally, the TVM should display clear instructions on how to seek help in case of an error.
- **Thirdly**, almost all the users confirmed that they would appreciate a feedback or a notification on their phones (either via email, text message or in-app-notification) so that they have some peace of mind having confirmed ticket and the transaction confirmation being sent on their phone. This will help users with the peace of mind and enable them to track their transactions and expenses.

In conclusion, interviews provide very valuable perception of the users on the design and operation of the TVM. The TVM needs to provide a clear and easily understandable user interface so that it can be accessed by any age group with any educational qualification and does not exclude anyone. By addressing these issues, TVMs can be more reliable, efficient, and over-all user-friendly experience for public.

Problem 4

4.1 Use case

A context of use is a description of the conditions under which a software system will be used in a normal working situation. In this project, those conditions are pertaining to the users, demands, and the physical, and environments in which the software system is used.

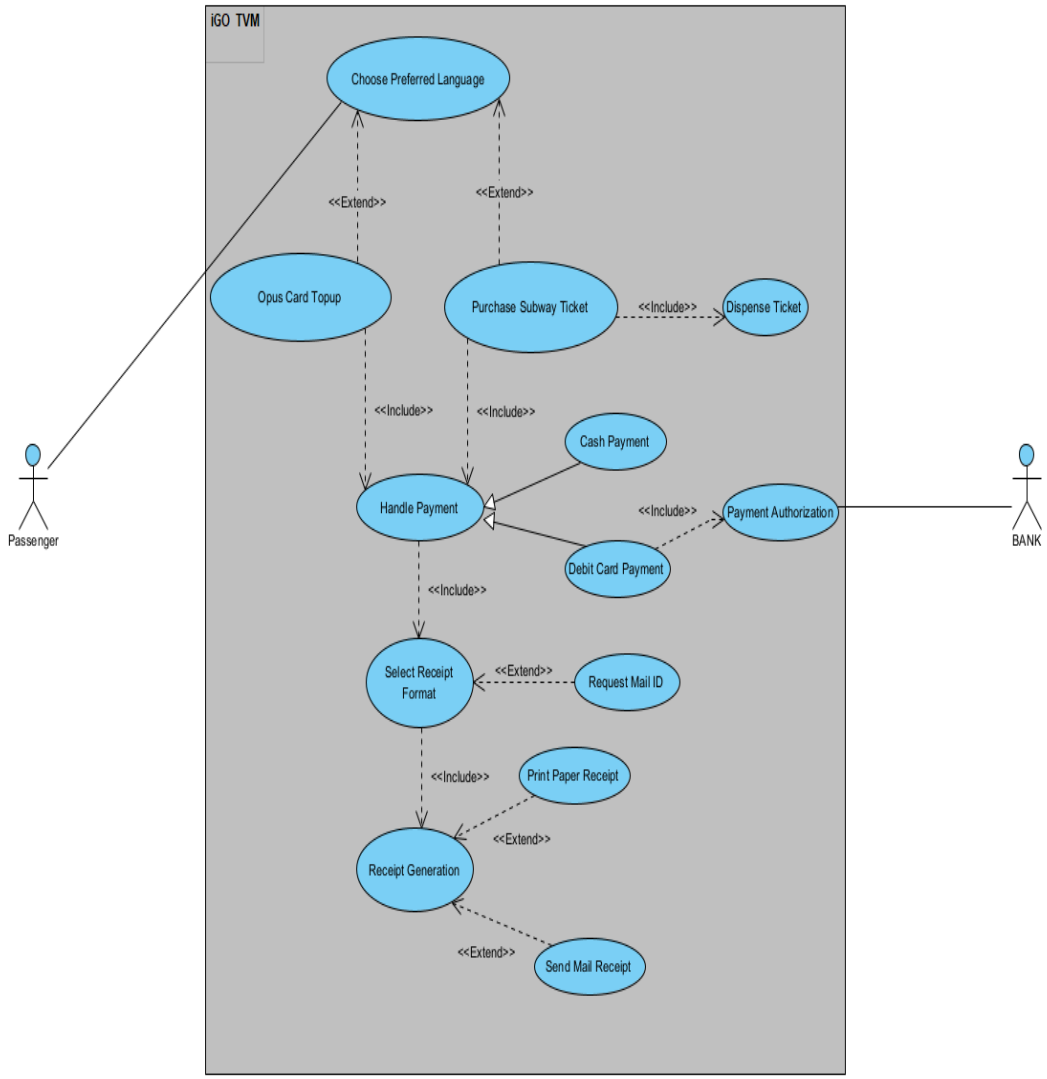

Figure 4.1: Use Case diagram

1. Choose Preferred Language :

Name	Choose Preferred Language
Description	Passenger should choose English or French
Actors	Passenger
Trigger	Passenger starts a session
Pre-Condition	The Passenger started a session
Post Condition	The language used in the message is set to the language selected by the Passenger
Normal Flow	• The Passenger chooses a languages • Then the Passenger recharges the Opus Card or purchases a subway ticket. • The transaction is processed. • A receipt would be generated.
Exception	The passenger cancels the session

2. Opus card Topup:

Name	Opus card Topup
Description	Passenger recharges the Opus Card using the iGO
Actors	Passenger
Trigger	Passenger selects the Opus card topup option.
Pre-Condition	The Passenger started a session and chosen the language
Post Condition	The Opus card recharge is done
Normal Flow	• The Passenger recharges the Opus Card • iGo TVM redirects to the Payment option.
Exception	The passenger cancels the session

3. Purchase Subway ticket:

Name	Purchase Subway ticket
Description	Passenger purchases ticket using the iGO
Actors	Passenger
Trigger	Passenger purchases ticket using the iGO
Pre-Condition	The Passenger started a session and chosen the language
Post Condition	The ticket is dispensed to the passenger
Normal Flow	• The Passenger chooses the ticket type • iGo TVM redirects to the Payment option.
Exception	The passenger cancels the session

4. Dispense Ticket:

Name	Dispense Ticket:
Description	iGO TVM prints a ticket for the user
Actors	iGo TVM
Trigger	When payment is successful
Pre-Condition	The Passenger started a session and chosen the language and ticket type and payment is made.
Post Condition	The Ticket is dispensed
Normal Flow	• The Passenger chooses the ticket type • iGo TVM redirects to the Payment option. • Payment is successful and the ticket is printed.
Exception	The passenger cancels the session

5. Handle Payment::

Name	Handle Payment:
Description	Passenger pays the amount based on the selection
Actors	Passenger
Trigger	Selecting either a recharge card or purchasing a subway ticket
Pre-Condition	Ticket type is selected or Recharge card is selected.
Post Condition	Payment is done successfully and a receipt is generated.
Normal Flow	• iGO TVM shows the options for payment. • Passenger chooses a payment option and makes the payment. • iGo TVM redirects to the preferred type of receipt page.
Exception	The passenger cancels the session or payment is rejected.

6. Select Receipt Type:

Name	Select Receipt Type:
Description	Passenger selects the preferred type of receipt, either through email or a physical copy.
Actors	Passenger
Trigger	Successful payment
Pre-Condition	Payment should be successful
Post Condition	The receipt is generated based on the user's preference.
Normal Flow	• The receipt preference menu is displayed to the user. • The passenger can select either by email or a physical copy.
Exception	Invalid Email

7. Generate Receipt:

Name	Receipt Generation:
Description	A receipt would be provided to the passenger.
Actors	iGO TVM
Trigger	Receipt preference is chosen.
Pre-Condition	Payment is successful and receipt preference is selected.
Post Condition	Passengers receive receipt based on their preference.
Normal Flow	• Passenger chooses the preferred type of receipt. • iGO TVM generates receipt based on the chosen receipt type.
Exception	

Problem 5

5.1 Description:

Software engineering uses UML activity diagrams, a form of behavior diagram, to depict the progression of operations and actions within a system. They are utilized to graphically illustrate the order of actions or processes involved in a certain task, along with the choices and circumstances that direct those operations. The passenger starts the activity with the use of iGo TVM. Then, The language of the application must be selected on the ticket vending machine. Thereafter, the passenger will have the option of recharging their opus card or buying a ticket. With the data, the computer will calculate the amount and offer options like cash or a debit/credit card. Once the payment is completed and authenticated by the bank, the user will get the receipt according to the option he selects email or paper.

5.2 Activity Diagrams

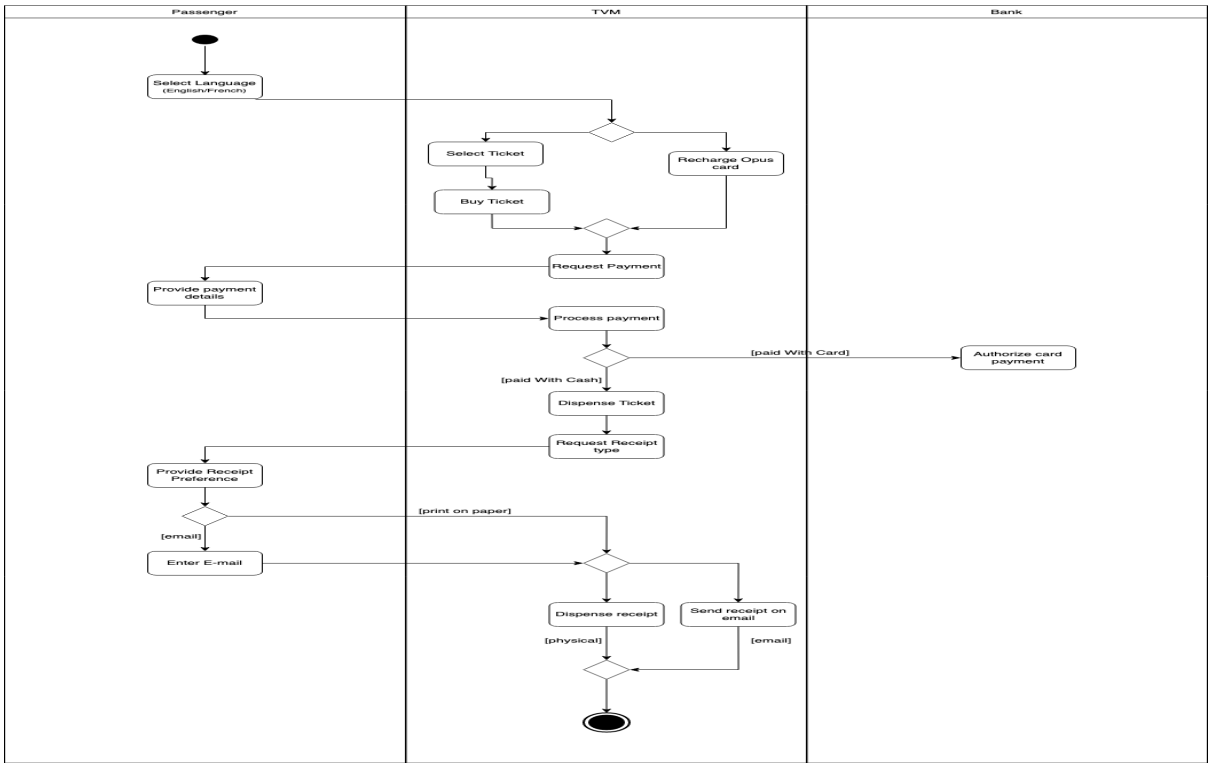

Figure 5.1: Activity Diagram by Deepanshu

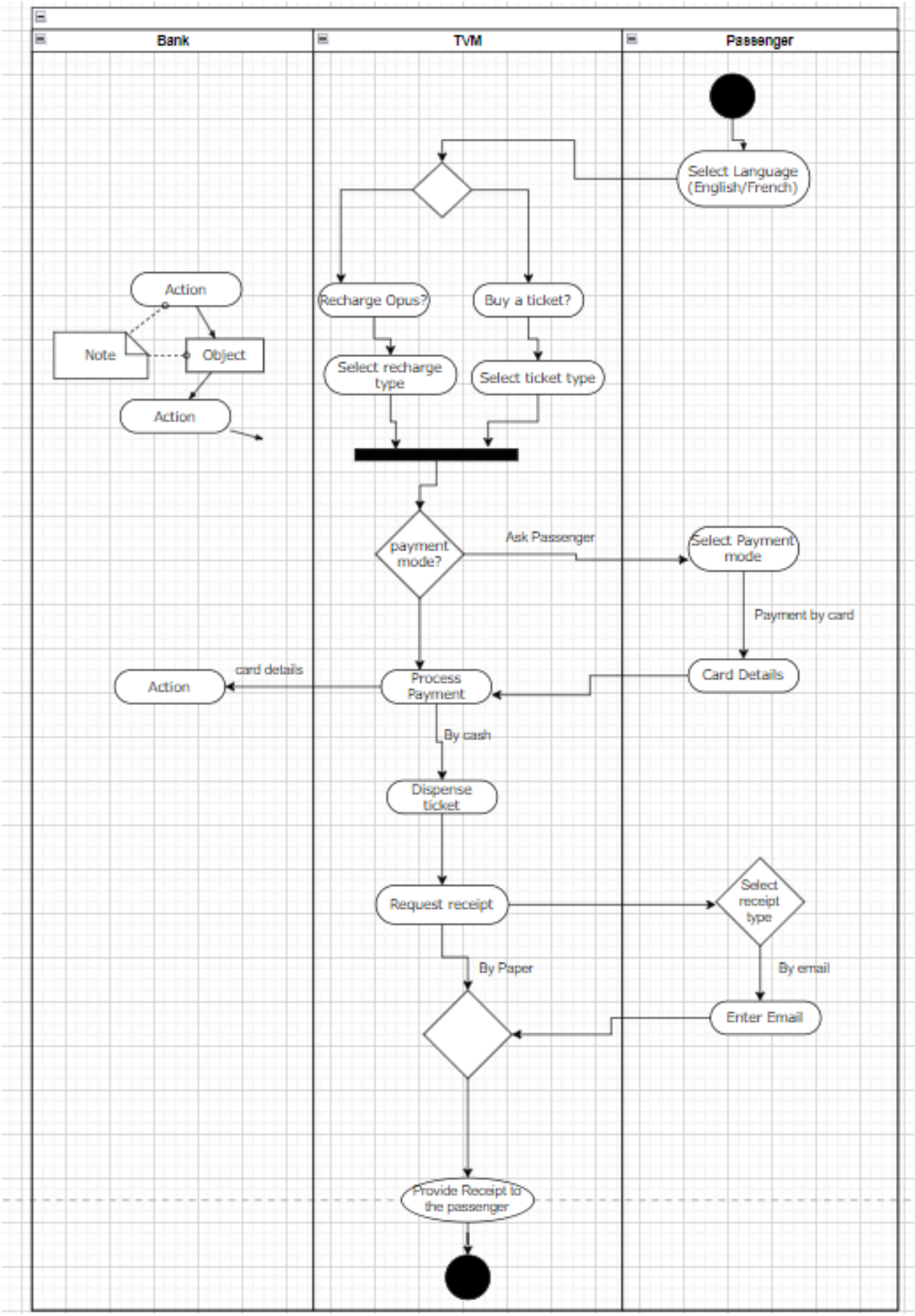

Figure 5.2: Activity Diagram by Saumya

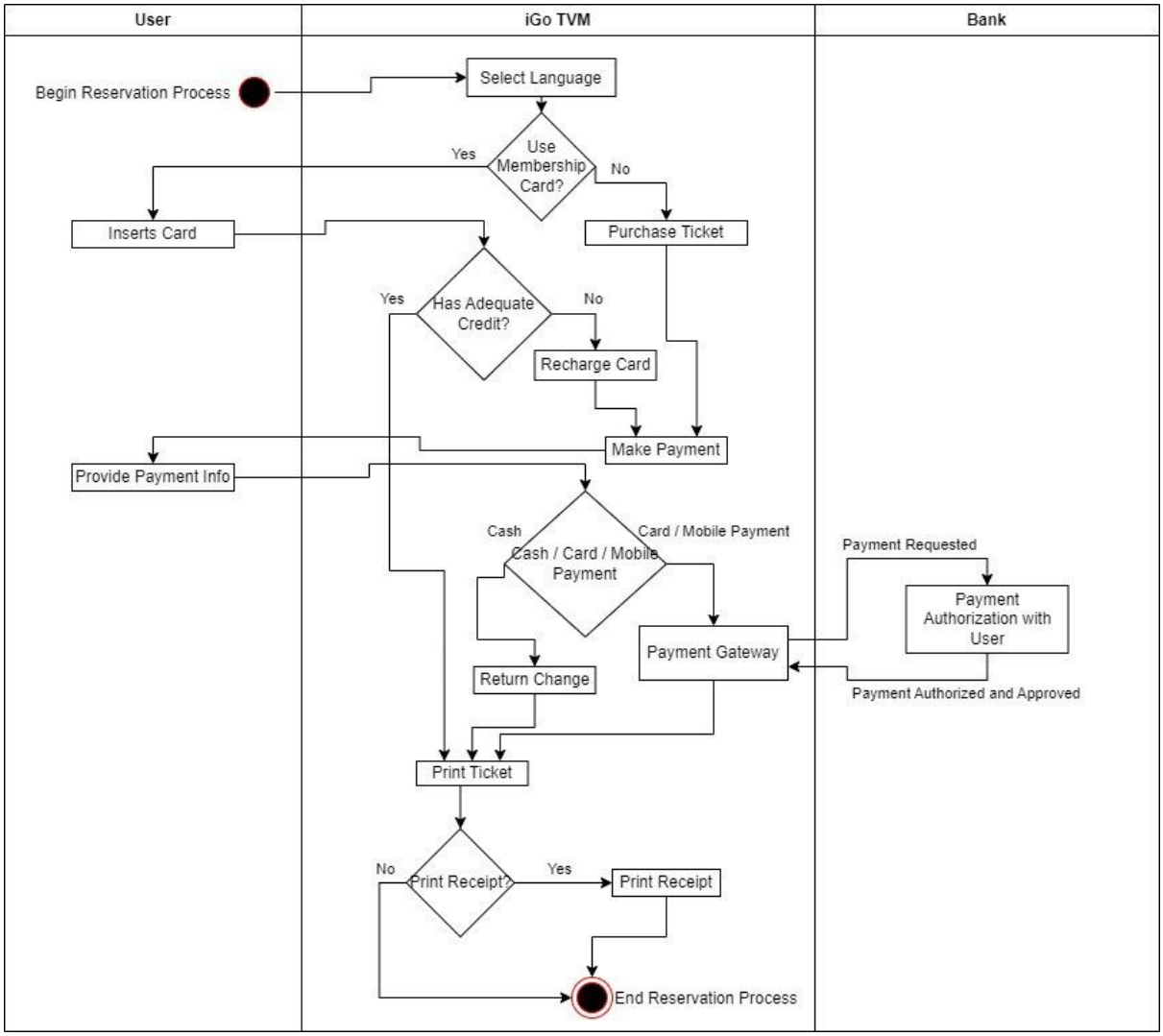

Figure 5.3: Activity Diagram by Apoorva

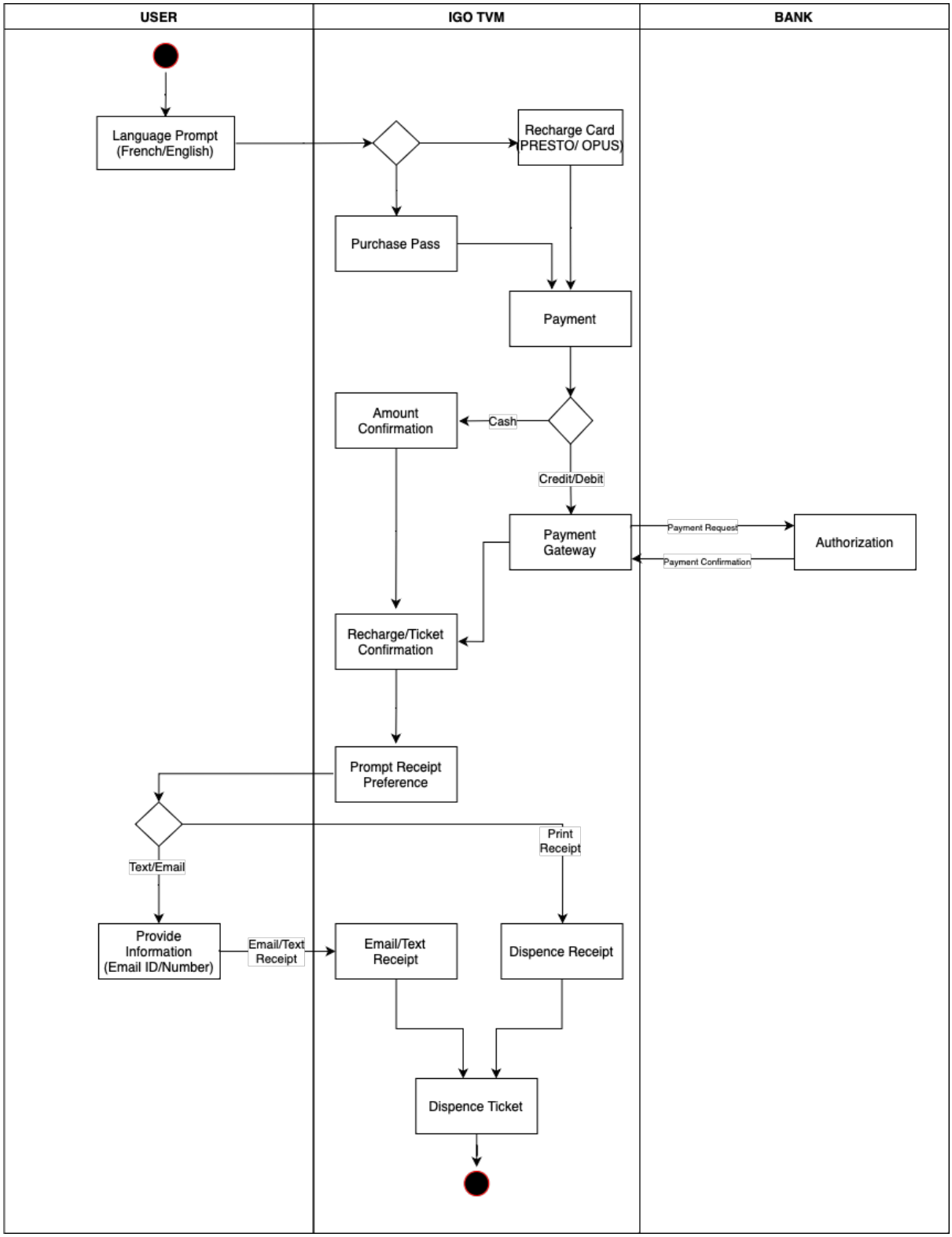

Figure 5.4: Activity Diagram by Aryan

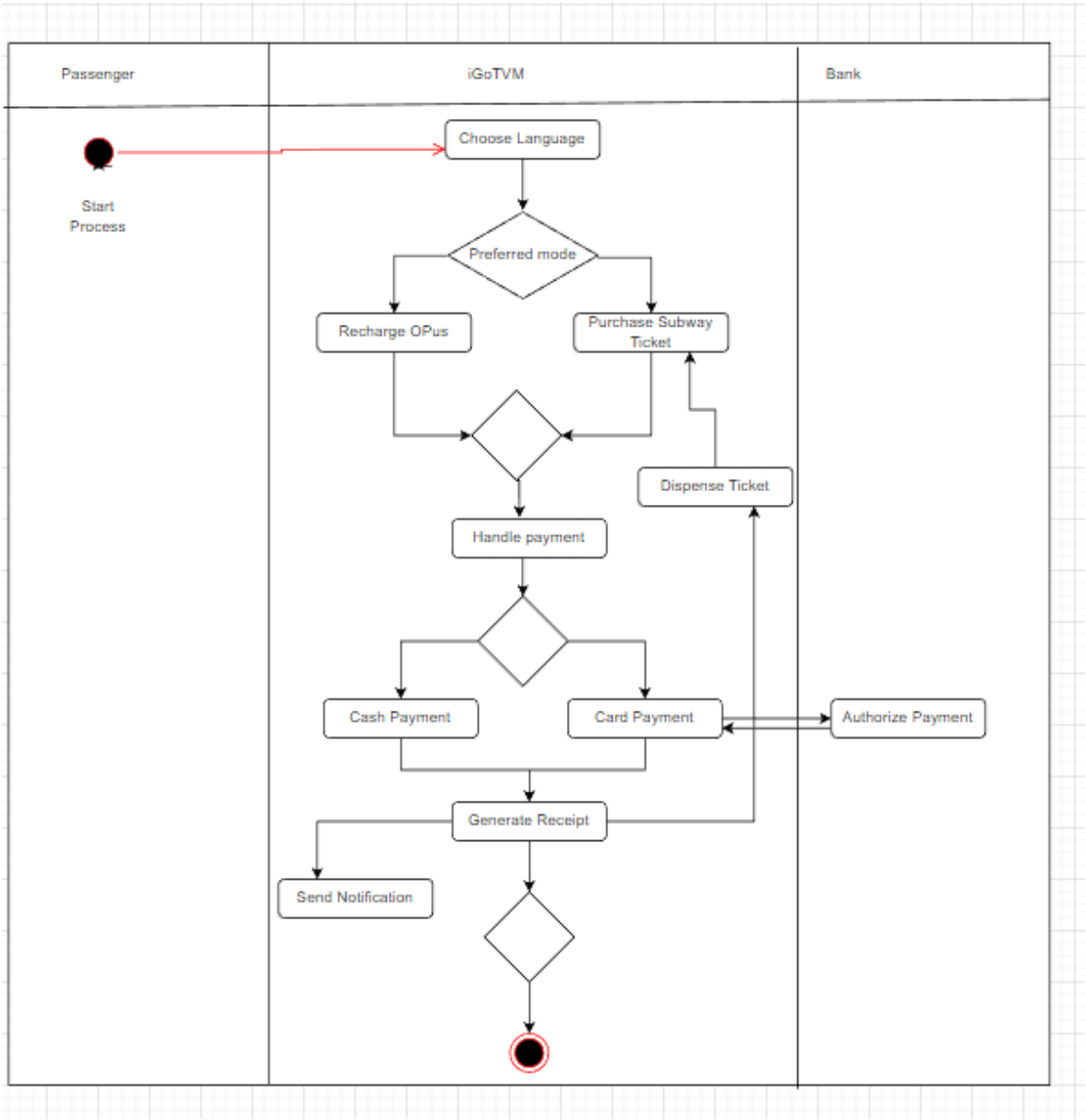

Figure 5.5: Activity Diagram by Shiva

Collaboration

- **Github:** `https://github.com/aryansaxena094/iGo.git`

Bibliography

[1] Higuchi, Yoshihiro. *History of the Development of Beverage: Vending Machine Technology in Japan.* National Museum of Nature and Science: Survey Reports on the Systemization of Technologies 7, 2007.

[2] Sandnes, Frode Eika, et al. *User interface design for public kiosks: an evaluation of the Taiwan high speed rail ticket vending machine.* Journal of information science and engineering 26.1, 2010: 307-321.