A logo of a university

Description automatically generated

American International University-Bangladesh (AIUB)  
**Department of Computer Science  
Faculty of Science &Technology (FST)  
Summer 22 23**

**<E-Payment-System-for-Transport>**

Software Requirement Engineering

Sec: **A**

Project submitted

By

**Alif Islam Siam (19-41486-3)**

**Adnan Sayed (19-41487-3)**

**Checked By Industry Personnel**

Name:

Designation:

Company:

Sign:

Date:

Software Requirements Specification

for

< E-Payment-System-for-Transport >

Version 1.0 approved

Prepared by-

Adnan Sayed (19-41487-3)

Alif Islam Siam (19-41486-3)

Most. Amena (20-43838-2) >

<27/8/2023>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction:

## Purpose:

The main purpose of the system is the online ticket payment where we are using 2-way system and there are some few scopes that creates the new interfaces for the stakeholders. The hassle- free online payment system creates the new identity for the passenger. Generation the QR code for an individual creates a new scope for the passengers and it opens a beneficial way for the stakeholders in the process. There are some few dependency’s of third party institutions and apps which creates a new policy to pay online which supports the system and the empowerment for each stakeholders.

## Document Conventions:

Descriptive titles are highlighted with bolding. Semi important terms are showed in italic manner. Different Versions will be released at a time.

## Intended Audience and Reading Suggestions:

This project is a prototype for the E-bus Ticket system. This has been implemented under the guidance of BRTA and Bangladesh mobile banking policy. This project is useful for the Bus owner, Driver of the bus, Passengers and as well as to the students.

## Product Scope:

## This project focuses on introducing a modern electronic payment method for public transportation, making it convenient for passengers to pay for their rides using digital options.

## Goals:

## Develop an easy-to-use digital payment system for transportation services.

## Enable passengers to pay via mobile apps, cards, or QR codes.

## Enhance passengers' travel experience by reducing payment hassles.

## Provide transportation authorities with better payment tracking.

***Key Components:***

**1. User-Friendly App:**

* Create a user app for smartphones both (Android and IOS).
* Allow passengers to set up accounts, log in securely, and manage payments.
* Provide balance checks and transaction history.
* Send notifications for account activities.

**2. System Infrastructure:**

* Integrate the payment system with secure online payment processors.
* Store user data, payments, and transactions securely.
* Implement strong access controls and data protection measures.

**3. Integration with Transport Services:**

* Collaborate with transportation companies to link the payment system with their services.
* Enable passengers to make payments via QR codes or contactless methods.

**4. Management Dashboard:**

* Develop a web-based dashboard for transportation staff to oversee the payment system.
* Manage user accounts, monitor transactions, and system health.

**5. Security and Compliance:**

* Implement robust security features to safeguard user information.
* Ensure compliance with data protection regulations.
* Integrate tools for fraud detection and prevention.

**6. Project Deliverables:**

* User-friendly mobile app (compatible with Android and iOS).
* Back-end system for processing payments and transactions.
* Integration with various transportation modes.
* Web-based dashboard for system management.
* User guides and training resources.

**7. Project Timeline:**

* Planning: 1 month
* Development: 4 months
* Testing and Refinement: 2 months
* Launch and Training: 1 month

**8. Assumptions:**

* Co-operation from transportation providers for integration.
* Complete budget and resources available.
* Being faithful to data protection regulations.
* Feasibility of integration with existing systems.

**9. Risks:**

* Challenges in integrating with diverse transportation technologies.
* User resistance to transitioning from cash to digital payments.
* Ensuring data security.

**10. Stakeholders:**

* Public transportation companies
* Commuters
* Development and Project Teams

**11. Budget:**

* The budget will be determined based on development costs, integration complexity, and maintenance needs.

## References:

1. Croock, M. S., & Taaban, R. A. (2021). Software engineering based secured E-payment system. International Journal of Electrical and Computer Engineering, 11(5), 4413.

1. Alhafi, R., Almutairi, S., Alsultan, N., Alsmadi, M. K., Alshabanah, M., Alrajhi, D., & Almarashdeh, I. (2019). E-Payment and Transactions using QR Codes. International Research Journal of Engineering and Technology, 6(2), 433-443.

1. Husni, E., & Hidayat, M. A. (2018, July). E-payment system using SMS gateway and line application. In 2018 International Conference on Information and Communication Technology for the Muslim World (ICT4M) (pp. 173-178). IEEE.

1. Al Farawn, A., Rjeib, H. D., Ali, N. S., & Al-Sadawi, B. (2020). Secured e-payment system based on automated authentication data and iterated salted hash algorithm. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(1), 538544.

1. O'mahony, D., Peirce, M., & Tewari, H. (1997). Electronic payment systems (pp. I-XII). Norwood: Artech House.

# Overall Description:

## Product Perspective:

E-payment system for transport is a digital payment system for paying bus fares.

• A user can pay cash less.

• No argument will occur with bus conductor for fares as there will be no conductor in the bus.

• Exact route fair with km calculations will be included in the software.

• If someone has less/no money, he/she can recharge within the app or can take emergency loan from the Fin-Tech Company which will be connected.

• Dedicated account will be there for different users.

• A user can pay using the QR code which will be present in every bus seats.

• Main software program will be written using C#.

• Software UI Design will be designed in Figma.

• Maps are used for designing transects, indicating observations and presenting results.

These maps are fetched by free map suppliers like Google maps.

• Fin-Tech Company like bKash will be integrated with our software.

• Route information will be taken from BRTA server.

A survey will be conducted with users before launching the software. A user will first open an account with necessary information. Then they’ll connect their bKash account. Then they can start paying the fares by scanning without any hassle. It’s that simple. Our app will be available only on Play store (Android).

## Product Functions:

## The main function of E - Bus ticket payment system is allowing the passengers to enter their data by scanning QR code and pay through online account. First every passenger has to enter the system by scanning the QR code than the identity information enters the system and check the validity of the passenger. After getting the individual information and identity, the system looks for the category of the particular passenger. If the passenger is the student then the discount price is added to the destination charge fee and generates. After paying the bill the system sends a confirmation code to the mobile operator. The payment has to done by any mobile operator. After confirming, the payment is added to the owners saving account direct through any mobile operator. The whole booking system is also checked and reconfirmed by the BRTA management authority by an email confirmation. However, if any passenger misses the drop-off location, the system will identify this and the passenger will receive an email and warning of the fault, and an extra charge will be imposed automatically, and the amount of the charge will be withdrawn from the passenger's mobile operator account.

## 2.3 User Classes and Characteristics:

|  |  |  |
| --- | --- | --- |
| User | Characteristics | Classes |
| Passengers | Add money  Emergency loan  Payment  Verified Account  Route check  Edit profile | Account  Profile  Route |
| Driver | Verified Payment  Profile Edit | Profile  Payment  Transaction History |
| Bus owner | Cash out money  Distribute Salary  Show bus info | Payment  Account  Transaction History  Bus profile |

## 2.4 Operating Environment:

Initially, the software application will be available for android devices only. But soon it will have IOS version also.

## 

## 2.5 Design and Implementation Constraints:

There are some issues that will limit the options available to the developers described below:

* **Strict Corporate, Regulatory Government Policies** - Strict regulatory laws can occasionally prevent software programmers from offering certain services that many users might find to be highly useful and practical. For instance, imagine that an engineer creates a GPS navigation application. The program would need the user's live location address, but there are some nations where accessing a citizen's location would violate that person's privacy. As a result, the application would not function without the user's location information. This might cause the creator to abandon the concept, which could have initially been incredibly beneficial to humanity in many ways.

* **Hardware Limitations** - Evidently, highly technological resources and equipment are needed to build sophisticated and effective applications. Hardware serves as the foundation for developers to create cutting-edge programs and apps. Therefore, it should be obvious that programmers would need new and cutting-edge technology. However, due to cost restrictions or a lack of the necessary technology, developers frequently lack these cutting-edge and effective tools while creating complicated programs. The availability of such enormous memory is a problem and a setback for the developer since some programs demand a lot of memory spaces in terms of computational memory, sometimes reaching up to a petabyte. Additionally, some complicated programs need a long time to compile before they can be assembled and run, making them time inefficient and useless to integrate without modern hardware like GPUs.
* **Interface to Other Applications**­­- The issues cannot be resolved by only creating the modules and programs. To produce the appropriate result, the Modules must be able to interact with one another and communicate with one another inside their own environment. Different applications must be able to connect to one another in a chain or link in order to interact and share information. However, many applications do not possess these qualities, and as a result, programs have trouble producing useful data and producing the desired outputs.
* **Language Requirements**- For developers, a programming language serves as their control panel. The choice of programming language is one of the most crucial factors for developers to think about when programming. Each programming language excels at certain tasks, whether it is JavaScript for creating websites or Python for creating video games. Similar to how technology is developing, new and effective coding languages must be considered in order to produce superior and sophisticated programs. To stay up with the demands of the clients, developers must keep themselves informed about the newest languages that have been developed. Our project developers have to use C# programming language and MYSQL as the project’s database.

* **Security Considerations**- The main goal and focus of any application must be to protect the user's data and keep it secret. However, their software is frequently compromised by skilled black hat hackers, who may acquire user information from these flawed programs, rendering their customers' sensitive information vulnerable to hacking assaults. Thus, software developers must constantly uphold the principle that their products must always be secure for consumers. To guard against security lapses and assaults, the program must be thoroughly and cautiously incorporated. As a result of dealing with data leak issues often, developers frequently deploy security patches for their programs.

## 2.6 User Documentation:

The application will be designed to be as simple to use as possible. Nonetheless, users may still require some supplementary information about each component of the system. The application will contain two features: The Tutorial and the Help menu.

The Help menu is a collection of topics covering each of the application's menus, features, etc. At any time, the user can navigate to the Help menu and select any of these topics to obtain more information.

The Tutorial takes all of these topics and condenses them into a single, step-by-step demonstration that the user can access immediately after installing the application. This tutorial is meant to quickly and effectively teach new users about the application.

## 2.7 Assumptions and Dependencies:

• Co-operation from transportation providers for integration.

• Complete budget and resources available.

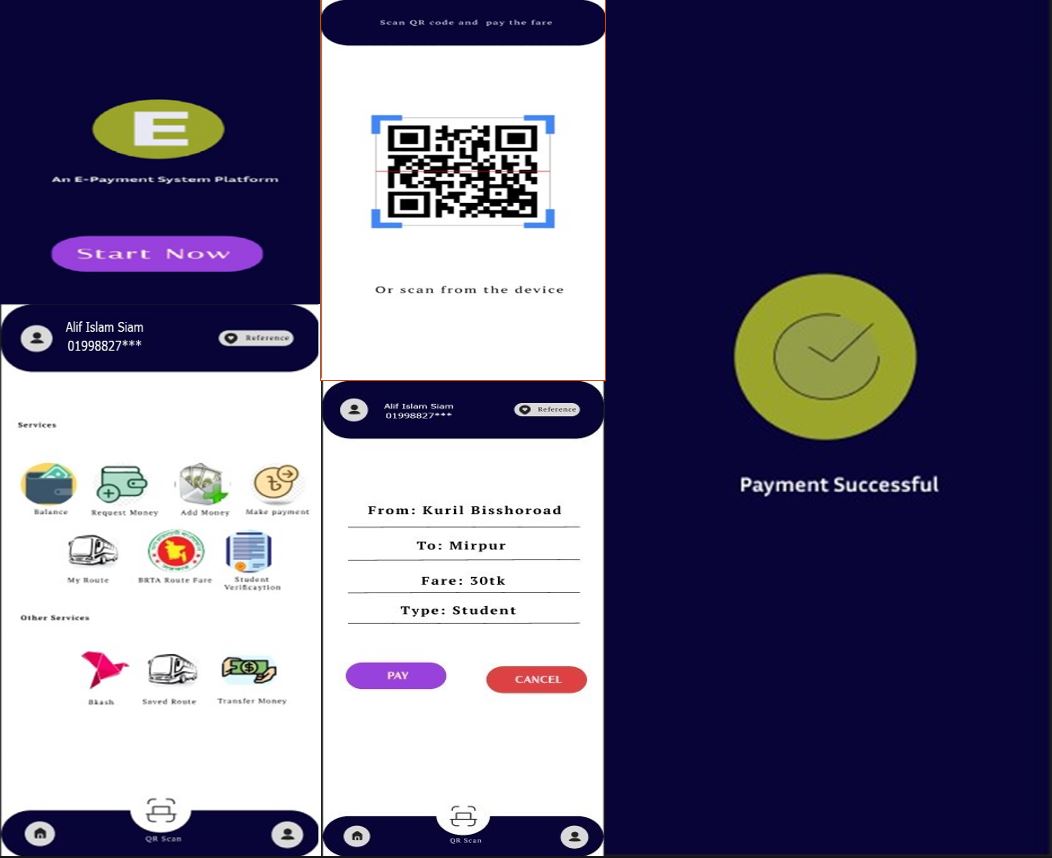
• being faithful to data protection regulations.

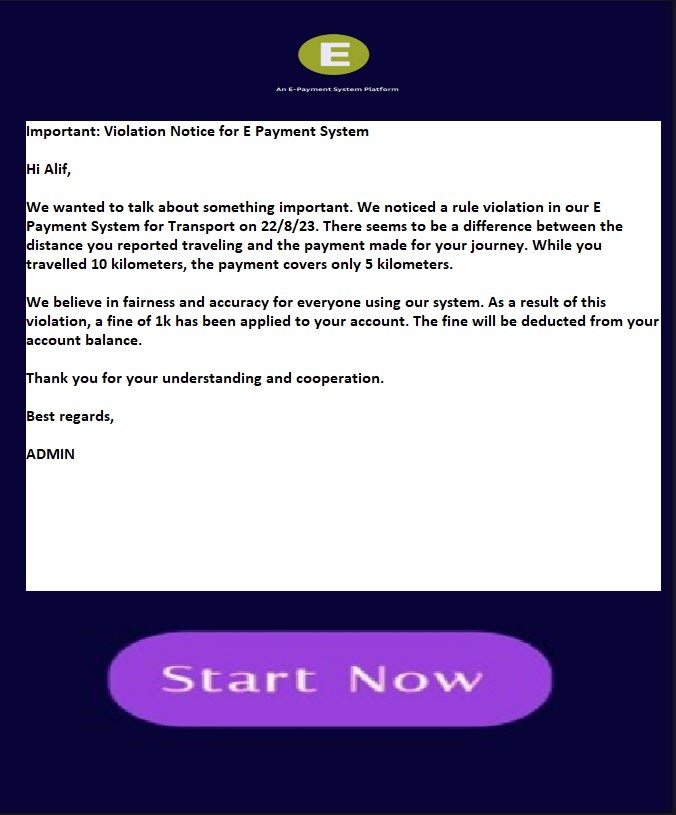
• Feasibility of integration with existing systems.

# External Interface Requirements:

## User Interfaces:

|  |  |
| --- | --- |
| **Screen** | **Details** |
| **Screen 1** | This is the starting page of the app. A user will tap into *“Get Started”* button to continue. |
| **Screen 2** | This will be the home page a user will get desired options to pay the fare.   1. **Balance-** Here user can see their total balance in their account. 2. **Request Money –** By this feature, a user can take emergency loan if they had low balance in their account. 3. **Add Money –** By this feature, a user can add money from their bKash Account direct to the payment app whenever they need. 4. **Make Payment –** By this feature, a user can scan the QR code and can make payment instantly. 5. **My route –** By this feature, a user can see the fares for different locations. 6. **BRTA Route fare –** This is the dedicated feature from BRTA which will help users to see the fares. 7. **Student verification –** If a user is a student then they can verify, Themselves by scanning their ID cards   8. **My bKash –** dedicated option from bKash to add/setup the account. |
| **Screen 3** | A user will scan the code and make payment**.** |
| **Screen 4** | Here details about the route and fare will be shown and user can pay instantly. |
| **Screen 5** | A successful message will be pop up and confirmation message will send to user’s mobile number |





## Hardware Interfaces:

**Card/Ticket Readers:**

* Devices where you tap or insert your payment card or digital ticket.
* Use for quick payments at entrances or on vehicles.

**QR Code Scanners:**

* Devices that read QR codes from your phone.
* Use to scan QR codes at terminals or on vehicles for payment.

**Mobile Apps (Smartphones):**

* Apps on your phone for making payments.
* Use to pay, check balance, and see transactions.

**Ticket Vending Machines:**

* Machines where you buy electronic tickets.
* Use for cash, card, or digital wallet payments.

**Admin Computers:**

* Computers used by staff to manage the system.
* Use to monitor transactions and generate reports.

**Displays:**

* Screens showing payment info and confirmations.
* Show what you're paying for.

**Biometric Scanners:**

* Devices using fingerprints or faces for security.
* Enhance safety and verify users

**Mobile Payment Devices:**

* Portable devices for payments on the go.
* Used to collect payments anywhere.

## Software Interfaces:

**User App Interface:**

* Users interact with the mobile app.
* Purpose: Users manage accounts, pay, and see transactions.

**Backend API Interfaces:**

* App talks to the system's backend.
* Purpose: Securely handle payments, user info, and history.

**Payment Processor Interface:**

* Backend talks to payment processors.
* Purpose: Ensure secure payment processing and authorization.

**Transportation System Interface:**

* Backend connects with transportation.
* Purpose: Share payment and entry info with transport.

**Admin Dashboard Interface:**

* Staff use a website to manage.
* Purpose: Monitor transactions, manage accounts.

**Database Interfaces:**

* Software interacts with the database.
* Purpose: Store and get user data, transactions.

**Security Interfaces:**

* Software layers connect with security tools.
* Purpose: Keep data safe, prevent fraud.

**Notification Interfaces:**

* System connects to send alerts.
* Purpose: Send messages, confirmations to users.

**Reporting Interfaces:**

* Interfaces to see summarized info.
* Purpose: Show transaction, system reports.

**Third-Party Integration Interfaces:**

* Connect to external services.
* Purpose: Blend external functions into the system.

**Authentication Interfaces:**

* Confirm user identity.
* Purpose: Verify logins, transactions.

**Error Handling Interfaces:**

* Deal with unexpected issues.
* Purpose: Manage errors, show friendly messages.

## Communications Interfaces:

**User App Talks to Backend:**

* App sends user info and payments to the system.
* Purpose: Users manage accounts and pay securely.

**Backend Talks to Payment Processors:**

* System sends payment details for processing.
* Purpose: Safely process and authorize payments.

**Backend Talks to Transport:**

* System shares payment data with transport services.
* Purpose: Allow entry after payment, link to transport.

**Backend Talks to Admin Dashboard:**

* System sends transaction updates to the dashboard.
* Purpose: Staff track transactions, manage the system.

**Backend Talks to Database:**

* System stores and gets data from the database.
* Purpose: Securely save user data and transactions.

**Backend Talks to Security Module:**

* System communicates with security tools.
* Purpose: Keep data safe, stop unauthorized access.

**Backend Talks to Notifications:**

* System sends messages and alerts to users.
* Purpose: Keep users informed about account actions.

**Backend Talks to Reporting:**

* System shares data for reports.
* Purpose: Help admins see trends and performance.

**Third-Party Connections:**

* System links to external tools.
* Purpose: Add extra functions using other services.

**User Verification:**

* System checks user identity.
* Purpose: Ensure secure access and transactions.

**Error Handling and Logging:**

* System shares errors for fixing.
* Purpose: Help improve the system, show friendly errors.

# System Features:

## System Feature:

**4.1.1 Description and Priority:**

Prioritization is a way to deal with competing demands for limited resources. When customer expectations are high and timelines are short, then we need to make sure the product delivers the most critical or valuable functionality as early as possible.

The requirement of successful prioritizations:

* The needs of the customers.
* The relative importance of requirements to the customers.
* The timing at which capabilities need to be delivered.
* Requirements that serve as predecessors for other requirements.
* Which requirements must be implemented as a group.
* The cost to satisfy each requirement.

There are four possible priority classifications for the requirements in a set. And these are: Must, Should, Could and Won’t.

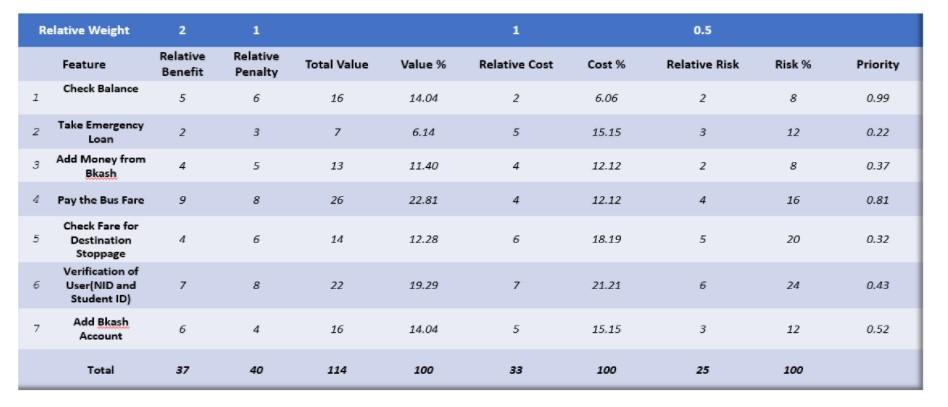
* Must: The requirement must be satisfied for the solution to be considered a success.
* Should: The requirement is important and should be included in the solution if possible, but it’s not mandatory to success.
* Could: It’s a desirable capability, but one that could be deferred or eliminated. Implement it only if time and resources permit.
* Won’t: This indicates a requirement that will not be implemented at this time but could be included in a future release.

There are three typical participants in the prioritization process and they are:

* The project manager or business analyst: They lead the process, arbitrate conflicts, and adjust prioritization data received from the other participants if necessary
* Customer: They representatives such as product champions, product managers, or product owners, who supply the benefit and penalty ratings.
* Development: They representatives who provide the cost and risk ratings.

There are six steps to use the prioritization model and these steps are described below:

1. List in the spreadsheet all the features, use cases, user stories, or functional requirements that we want to prioritize against each other.
2. Have the customer representatives estimate the relative benefit each feature would provide to the customer or to the business on a scale.
3. Estimate the relative penalty that the customer or the business would suffer if each feature were not included.
4. Calculates the total value for each feature as the sum of its benefit and penalty scores.
5. Developers have to estimate the relative cost of implementing each feature.
6. Developers rate the relative technical risk associated with each feature. The spreadsheet will calculate the percentage of the total risk that comes from each feature.



# Other Nonfunctional Requirements:

Since our project is call e-payment system and it will be used as an on the go application for all kinds of passengers and the passengers use this app for hassle-free payment to public transport so it has some performance requirement.

## Performance Requirements:

Must connected with high-speed internet.

* **Limit:** The framework ought to have the option to oblige Upto 5000 clients.
* **Reaction time**: Response season of the framework ought to be exceptionally low.
* **Normal reaction time:** 3-6 seconds.
* **Greatest reaction time:** 10-15seconds.

## Safety Requirements:

Safety issues that might occur.

* + - * Identity theft
      * Data breaches
      * Malware and viruses
      * Fake bills
      * Online scams
      * Faulty privacy settings

Safety measures that should be taken.

* + 1. **Secure internet connection:**

It is necessary to have access to a secure internet connection in order to pay the transport bill online. There should be a prohibition on the use of public Wi-Fi (if available) in order to prevent data theft.

* + 1. **Choose strong passwords:**

Passwords are one of the biggest cybersecurity weaknesses. People often choose passwords that are easy to remember – and, therefore, easy for hackers to guess. Select strong passwords that are harder for cybercriminals to demystify.

* + 1. **Enable multi-factor authentication where you can:**

Multifactor authentication (MFA) is an authentication method that asks users to provide two or more verification methods to access an online account. For example: Fingerprint / Face unlock / Password.

* + 1. **Keep the app up to date:**

App should be updated at regular basis because developers are constantly working to make products safe, monitoring the latest threats and rolling out security patches in case of vulnerabilities. By using the latest versions of your operating systems and apps, you benefit from the latest security patches. This is especially important for apps that contain payment etc.

* + 1. **Review your privacy settings and understand privacy policies:**

Privacy policies provided by the company should be followed before using the app. Privacy settings should be reviewed for better security of the individual user account.

## Security Requirements:

The User's Information must be kept confidential and should only be available to him or her. Sharing of information should take place in a secure setting where it cannot be hacked by an outsider and stored. End-to-end encryption is a better solution to meet the security requirement.

**Administrator comfort level:** The administrator of the framework will have a minimum of three days to prepare for the framework. The framework's features should all be obvious to users. The framework's throughput must be highly adequate in order to provide clients with ongoing assistance.

## Software Quality Attributes:

Quality of a software may vary from others. Different perspectives can be taken into consideration when defining quality.

The following factors are used to measure Software Development Quality. Each attribute can be used to measure product performance. These attributes can be used for Quality assurance as well as Quality control.

* **Reliability:**

By this quality, a software’s consistency is measured in certain situations. Whether it is reliable to the user or not.

* **Maintainability:**

Software should be easy to handle and maintain for future development. A developer should be able to add the code to the existing system code to upgrade/add new features on time basis.

* **Usability:**

A system should be fully user friendly. It should be easy to use, learn and also navigate because it’ll be use by all sorts of people including uneducated and older people.

* **Portability:**

A system should be portable one in terms of costing and technical issues.

* **Correctness:**

The application should be correct in terms of its functionality, calculations used internally and the navigation should be correct. This means that the application should adhere to functional requirements.

* **Efficiency:**

This quality is one of the major system quality. Main idea of making this system is to pay the fair instantly without any hassle and in less time. A system should utilize the system’s processor and ram to make the system and usability fast.

* **Integrity or Security:**

Integrity comes with security. System integrity or security should be sufficient to prevent unauthorized access to system functions, prevent information loss, ensure that the software is protected from virus infection, and protect the privacy of data entered into the system.

* **Testability:**

The system should be easy to test and find defects. If required, it should be easy to divide into different modules for testing.

* **Flexibility:**

Should be flexible enough to modify. Adaptable to other products with which it needs interaction. Should be easy to interface with other standard 3rd party components.

## Business Rules:

As the name suggests, a business rule is a rule that defines a constraint within the context of a business that is specific to that business. Similarly,

Our system has also some certain business rules which are to be followed. These are stated below;

* A user will have to open an account using their NID card.
* Contact number will be needed to register.
* A user will have to enter strong password for better security.
* Proper security measures will be ensured using multifactor authentication.
* A user can add their picture as a profile picture.
* A user to verify their fingerprint before using it.
* After completing all the security measures, user have to connect their e-wallet account such as bKash.
* A user will have to pay by providing their details and destinations.
* If the user is a student, they have to register by using an option “I am a student”.
* A token or a receipt will be provided after each payment.

# Other Requirements:

6.1 Appendix A: Glossary:

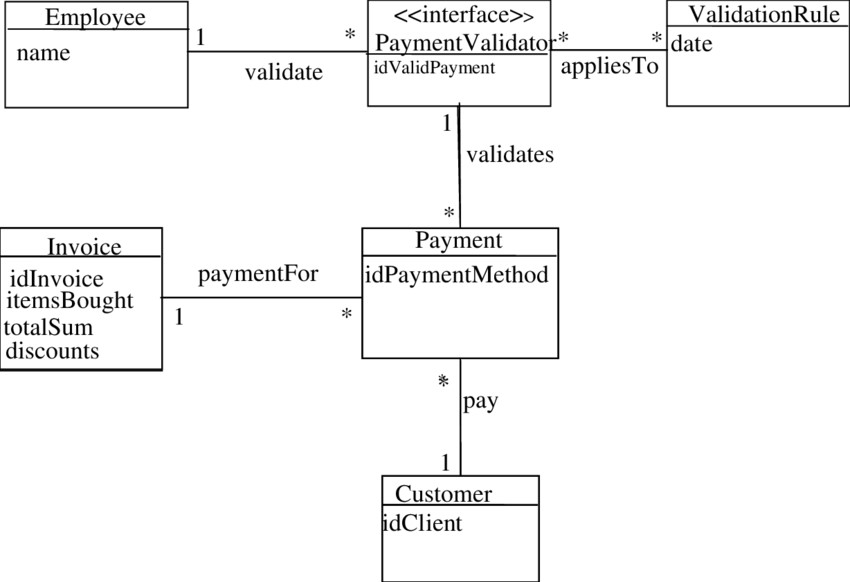
Terms:

1. Passengers only have to use phones or cards to make payment for transportation service.
2. Paying the fare in online not with cash.
3. A confirmation code that holds information is frequently used in payment systems to send payment information quickly and securely.
4. Every user can store their information and payment details securely.
5. Payment Processor is a secure service that manages payment transactions, ensuring that payment is delivered securely between the payer and the recipient.
6. Users interact individually to the user interface through the app to making payments
7. Backend system manage data including user accounts, transactions, security.
8. Verify the identity of users before allowing them access to their account or making payments.
9. API (Application Programming Interface) allows different software systems to communicate and interact with each other. Example: Users BKash account connected to the E-Payment for transport system for making payment.
10. Message will sent to users about any update, events related to payment.
11. Data will encrypted to code to prevent unauthorized access
12. Instructions provided to users to help them understand how to use the payment system effectively.
13. Legal guidelines to dictate user data collection.

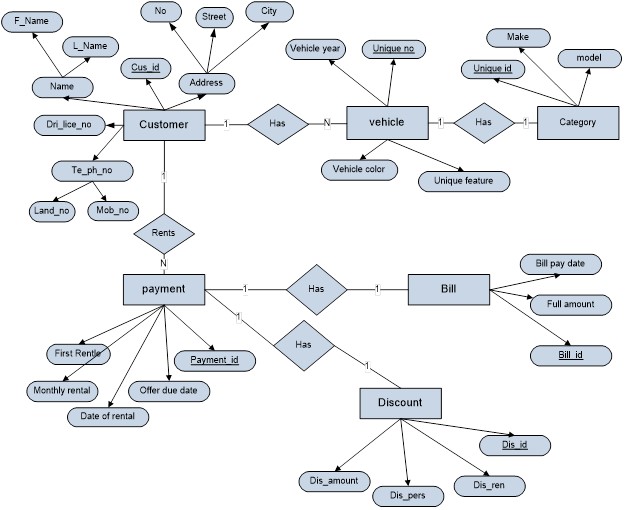
* **Abbreviations:**
* **QR code**- Quick Response Code, a type of two-dimensional barcode that can be scanned by smartphones and other devices to quickly access information.
* **BRTA-** Bangladesh Road Transport Authority, the government agency responsible for regulating road transport and traffic in Bangladesh.
* **IOS-** iPhone Operating system developed by Apple Inc. for their mobile devices like iPhones and iPads.
* **Fin-Tech Company-** Financial Technology Company, a company that leverages technology to provide financial services and solutions. Like bKash.
* **Figma-** A cloud-based design tool used for user interface and user experience design collaboration.
* **GPS navigation-** Global Positioning System navigation, a technology that provides real-time location and navigation information using satellites.
* **GPUs-** Graphics Processing Units, specialized hardware used for rendering graphics and performing complex calculations, commonly used in computer graphics and machine learning.
* **bKash-** A popular mobile financial service of Bangladesh that allows users to send and receive money, pay bills, and more through their mobile phones.
* **NID card-** National Identity Card, a government-issued identification card that is used to establish a person's identity.

6.2 Appendix B: Analysis Models:

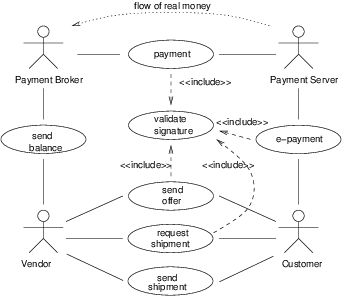
* **Class Diagram:**



* **ER Diagram:**



* **Use Case Diagram:**



6.3 Appendix C: To Be Determined List

|  |  |
| --- | --- |
| **Features** | **Description** |
| **Data Backup and Recovery** | Data backup & recovery system will be added in future. As, Users data will be recorded in the software , data backup and the process of data recovery system will help to find users data in case of system failures. |
| **Promoting the system** | Online payment in Transport is a hassle free payment system for users. At the beginning, few people will know about the payment system but with enough promotion, everyone will be able to use it to make payments without difficulty. |
| **Web Interface Design** | The design of the web interface for the e-payment system will include customer experience. |
| **Strong security** | Determine the level of security required to protect sensitive data, including passwords, credit card numbers, and other personal information. This might include access controls, secure servers, and encryption techniques |
| **Detecting & preventing Fraud** | Unauthorized access will be prevent for making payment or using the system. |