American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

E-Payment System For Transport

A Software Requirement Engineering Project Submitted
By

Semester: Spring_22_23		**Section: H**	**Group Number:4**	
SN	Student Name	Student ID	Contribution in % (CO+CO2)	Individual Marks
1	Nipa, Nilima Rani Kar	20-43009-1		
2	Athoy Kanti Ray	20-43259-1		
3	Muqeet Ryan	18-36653-1		
4	A.M.Nazmul Hossain	20-43037-1		

The project will be Evaluated for the following Course Outcomes

Evaluation Criteria	Total Marks (50)	
Introduction, Format, Submission, Defense	[10 Marks]	
System Overall Description & Functional Requirements	[10 Marks]	
System Quality Attributes and Project Requirements	[10 Marks]	
UML and E-R Diagram with Data Dictionary	[10 Marks]	
UI/UX Prototyping	[10 Marks]	

Software-Requirements Specification

for

E-Payment-System-for Transport

Version 1.0 approved

Prepared by

Nilima Rani Kar Nipa (20-43009-1)

Athoy Kanti Ray (20-43259-1)

Muqeet Ryan (18-36653-1)

A.M.Nazmul Hossain (20-43037-1)

<02/05/23>

Table of Contents

1. Introduction

1.1 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 interface s for the stack holders. 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 stack holders 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 stack holders.

1.2 Document Conventions

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

1.3 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, conductor of bus, Passengers and as well as to the students.

1.4 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.

2. 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.

3. 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.

4. 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), 538-544.

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

2. Overall Description

2.1 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.
- Exact route fair with km calculations will be included in the software.
- If some one has less/no money, he/she can recharge within the app or can take emergency loan from the FinTech 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.
- FinTech 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).

.

2.2 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. Effectively after the confirmation code sent, the contractor has to recheck the payment system by scanning the QR code system. After reconfirming 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.

2.3 User Classes and Characteristics

For Different user have different classes. In this software there are three types of users.

User	Characteristics	Classes
Passengers	Add money Emergency loan Payment Verified Account Route check Edit profile	Account Profile Route
Conductor	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 it's IOS version also for iPhone users.

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.

3. System Requirements

3.1 System Features

3.1.1 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.

A prioritization matrix for this system is described below:

	Relative Weight	2	1			1		0.5		
	Feature	Relative Benefit	Relative Penalty	Total Value	Value %	Relative Cost	Cost %	Relative Risk	Risk %	Priority
1	Check Balance	5	6	16	14.04	2	6.06	2	8	0.99
2	Take Emergency Loan	2	3	7	6.14	5	15.15	3	12	0.22
3	Add Money from Bkash	4	5	13	11.40	4	12.12	2	8	0.37
4	Pay the Bus Fare	9	8	26	22.81	4	12.12	4	16	0.81
5	Check Fare for Destination Stoppage	4	6	14	12.28	6	18.19	5	20	0.32
6	Verification of User(NID and Student ID)	7	8	22	19.29	7	21.21	6	24	0.43
7	Add Bkash Account	6	4	16	14.04	5	15.15	3	12	0.52
	Total	37	40	114	100	33	100	25	100	

3.2 Non-Functional/Quality 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.

3.2.1 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.

3.2.2 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.

I. **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 WiFi (if available) in order to prevent data theft.

II. **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.

III. **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

IV. 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.

V. 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.

3.2.3 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.

3.2.4 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.

I. Reliability:

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

II. 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.

III. 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.

IV. Portability:

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

V. 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.

VI. 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.

VII. 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.

VIII. Testability:

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

IX. 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.

3.2.5 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.

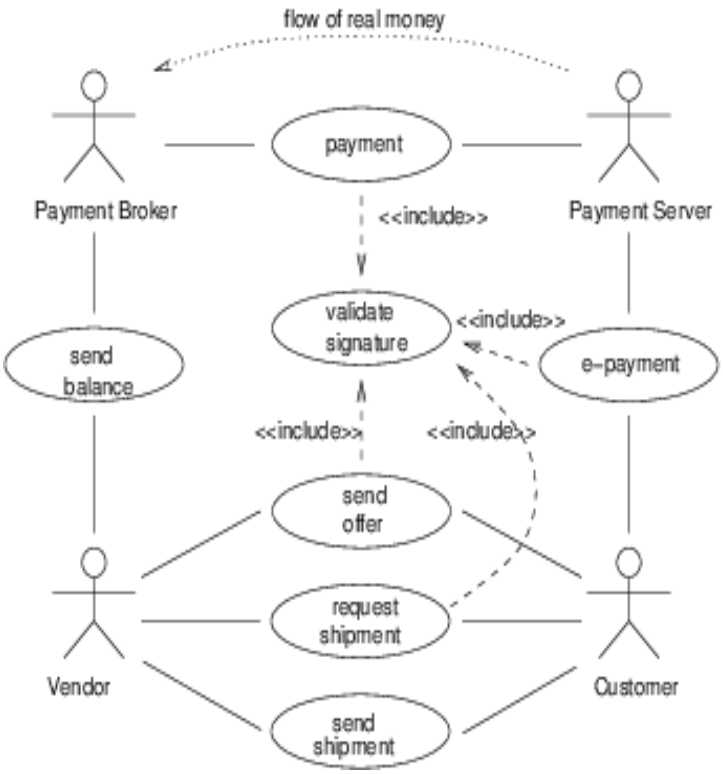
3.3 Project Requirements

- **Tools:** Used selenium tools for this project.

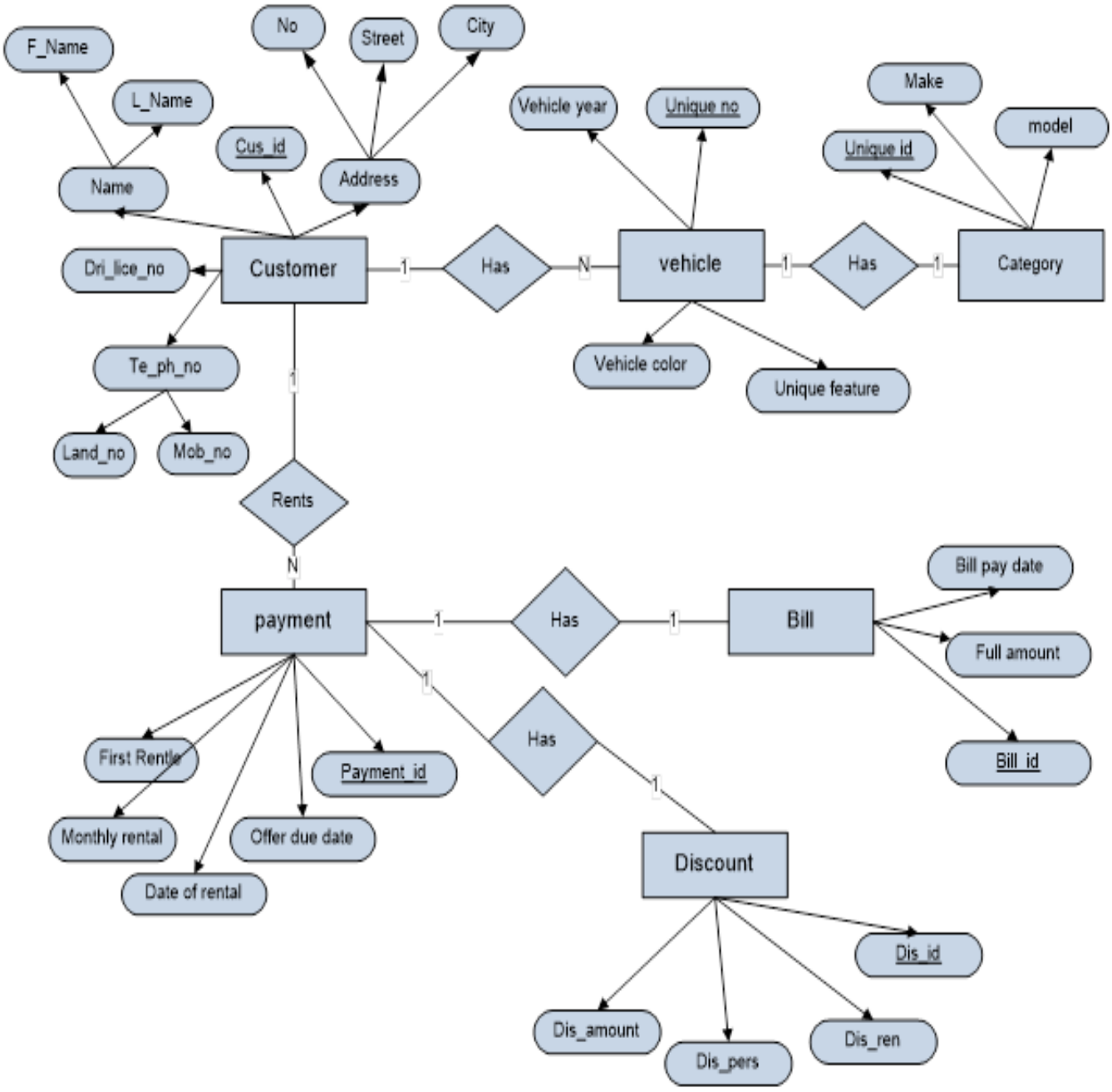
4. Design and Interface Requirements

4.1 UML Diagrams

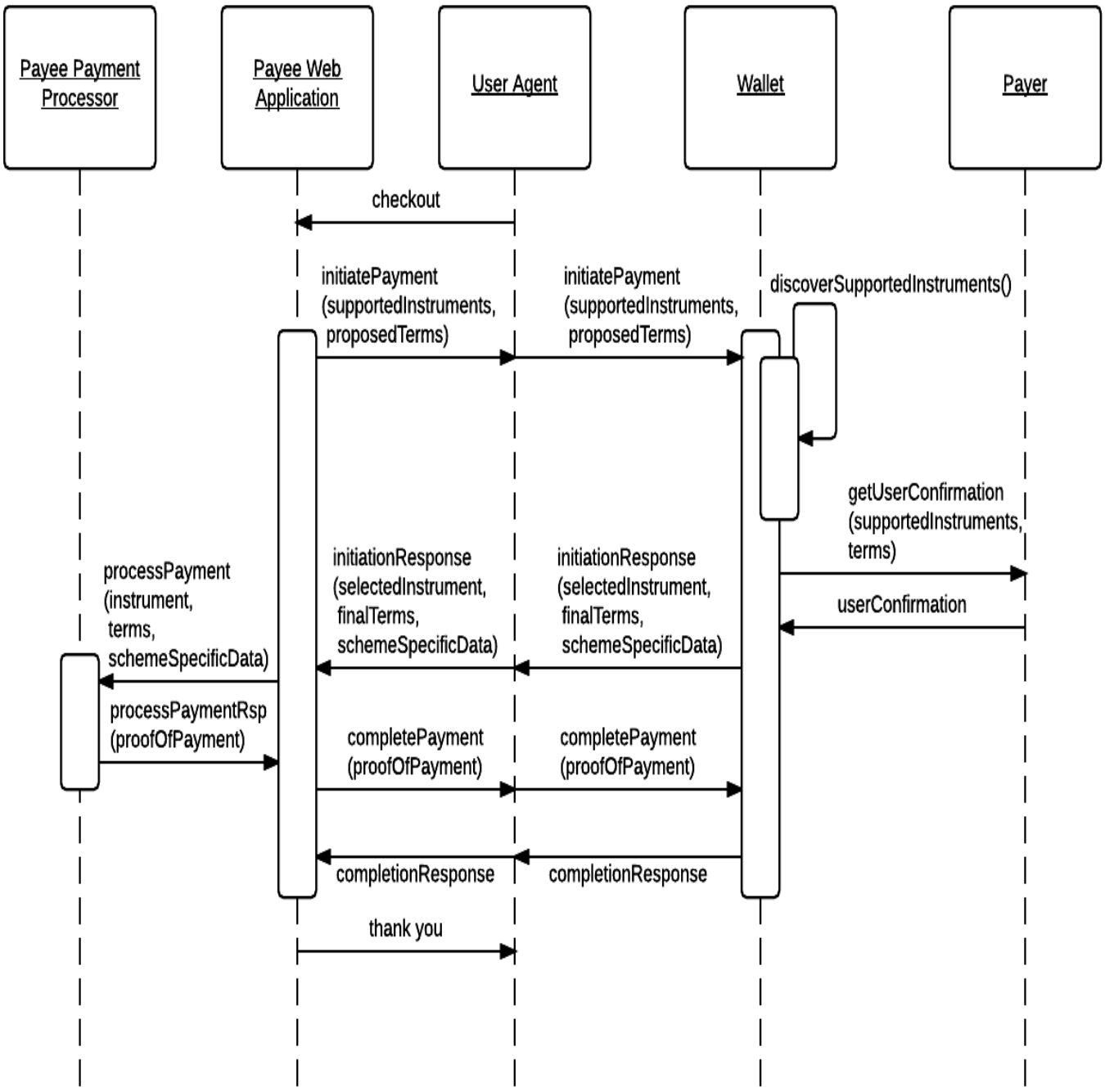
Use Case Diagram:



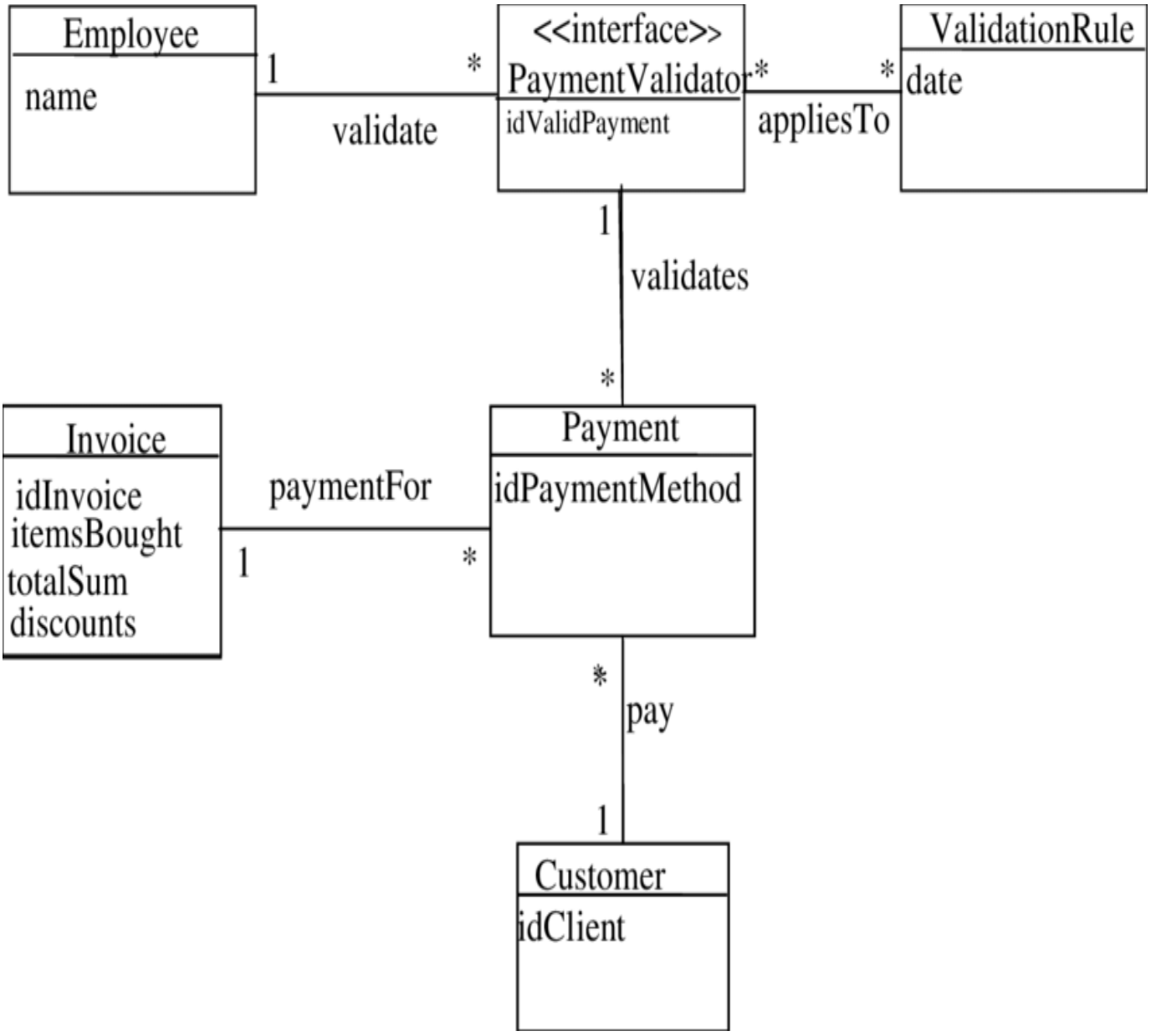
ER Diagram:



Sequence Diagram :


Payee Payment Processor
Payee Web Application
User Agent
Wallet
Payer
checkout
initiatePayment (supportedInstruments, proposedTerms)
initiatePayment (supportedInstruments, proposedTerms)
discoverSupportedInstruments()
getUserConfirmation (supportedInstruments, terms)
userConfirmation
processPayment (instrument, terms, schemeSpecificData)
initiationResponse (selectedInstrument, finalTerms, schemeSpecificData)
initiationResponse (selectedInstrument, finalTerms, schemeSpecificData)
processPaymentRsp (proofOfPayment)
completePayment (proofOfPayment)
completePayment (proofOfPayment)
completionResponse
completionResponse
thank you

Class Diagram:



4.2 Data Dictionary

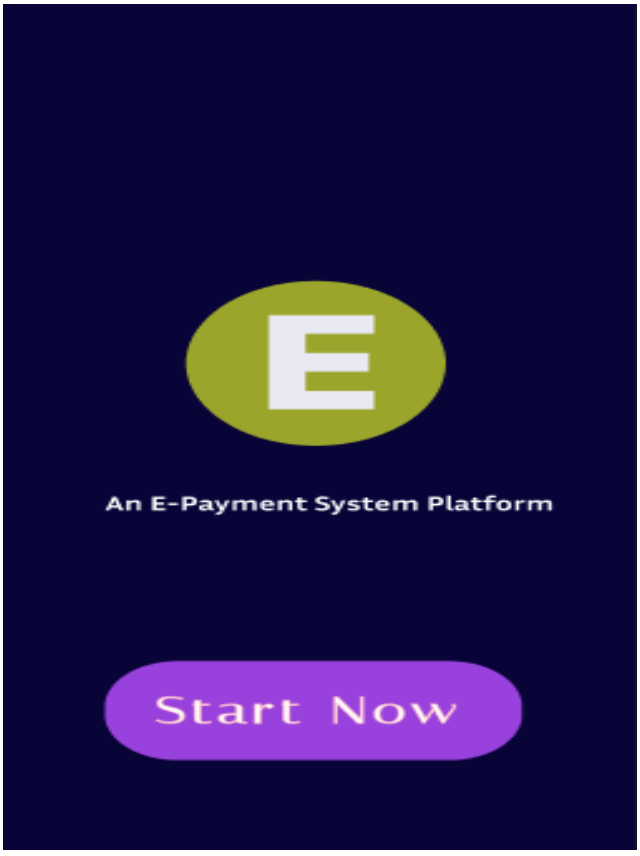
Entity	Attribute	Type/Size	Validation	Key
Pupil	PupilID	Number (5)	10000-99999	Primary
Pupil	Forename	Text (10)	Required	
Pupil	Surname	Text (15)	Required	
Pupil	DOB	Date (8)	Valid Date	
Teacher	TeacherID	Number (3)	100-999	Primary
Teacher	Name	Text (15)	Required	
Teacher	Room	Text (3)		
Teacher	Subject	Text (10)		

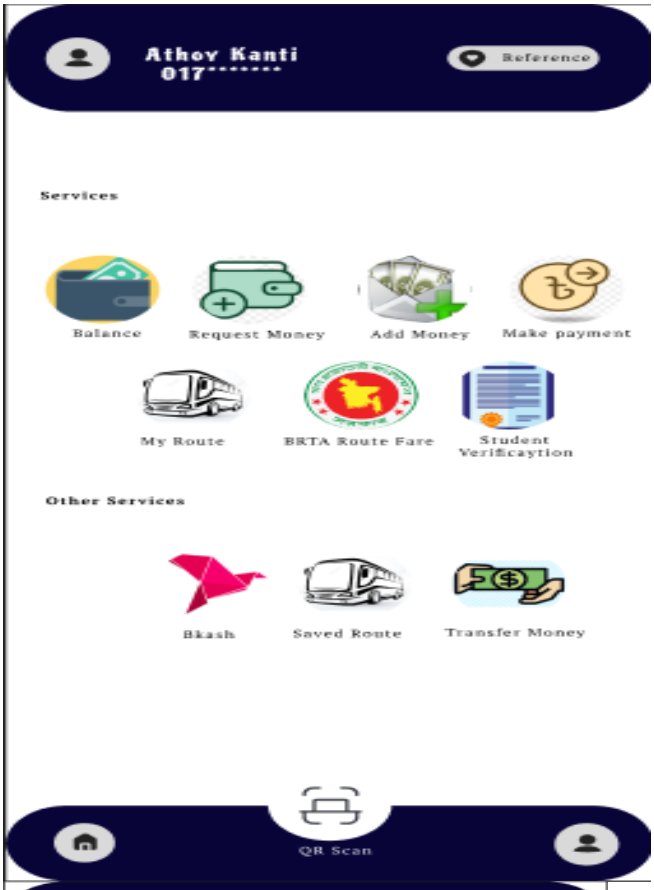
4.3 UI/UX Design Specification

User Interfaces :

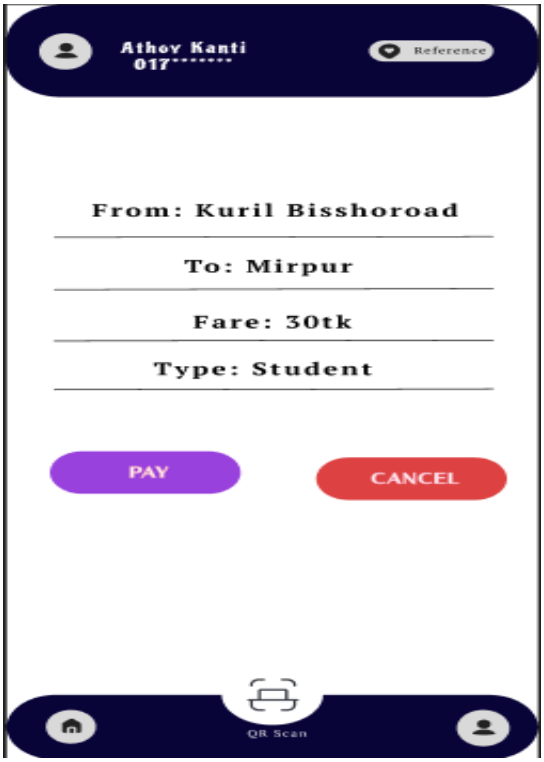
Screen	Details
Screen 1	This is the starting page of the app. A user will tap into *"Get Started"* button to continue.
Screen 2	This will be the home page a user will get desired options to pay the fare. 1. **Balance-** Here user can see their total balance in their account. 2. **Request Money –** By 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

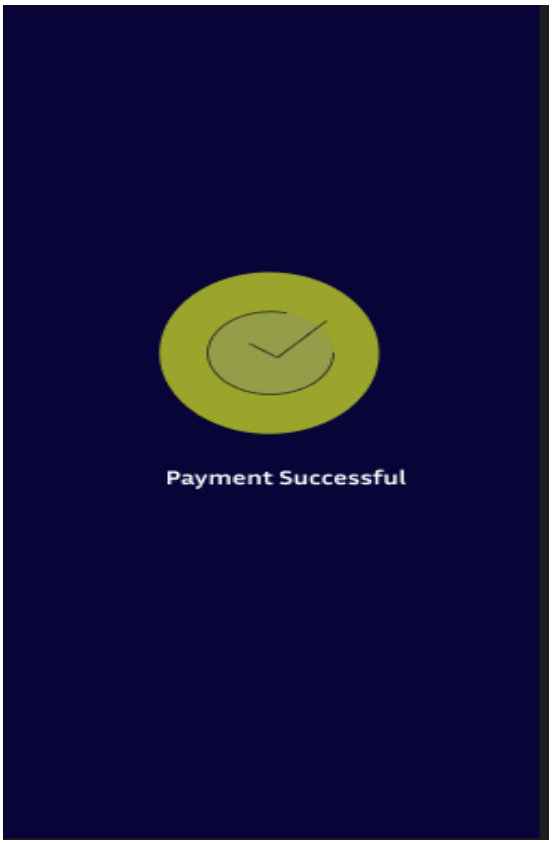
	themselves by scanning their ID cards. 8. **My Bkash –** dedicated option from Bkash to add/setup the account.
Screen 3	A user will scan the code and make payment**.**
Screen 4	Here details about the route and fare will be shown and user can pay instantly.
Screen 5	A successful message will be pop up and confirmation message will send to user's mobile number.




Athoy Kanti
017********
Reference
Services
Balance
Request Money
Add Money
Make payment
My Route
BRTA Route Fare
Student Verificaytion
Other Services
Bkash
Saved Route
Transfer Money
QR Scan


Scan QR code and pay the fare
Or scan from the device


Athoy Kanti
017********
Reference
From: Kuril Bisshoroad
To: Mirpur
Fare: 30tk
Type: Student
PAY
CANCEL
QR Scan


Payment Successful