Dhaka Digital Metro Railway Ticket System

The system will help us to buy ticket with our smart NID card which will be faster transaction .So there will be no line and this will help the people to save time and also makes our life easier.

Justifications: Bangladesh is a densely populated country. Near 21,006,000 people live in Dhaka. For this people we need a high speed transportation. That's why the metro railway system came. For high speed transportation we also need a high speed ticket booking system. There The Dhaka Digital Metro Railway Ticket System comes. This ticket booking system will help people buy tickets from station booth fast .so, there will be no line to stand for tickets. This will safe time for the passenger .There are smart ticket booking system in India, china and many other countries. But there is a drawback they have to buy Special smart card to use the smart ticket booking system. For this many people don't buy smart card and this why the use of smart ticket booth is really poor .And people don't get the benefit of smart ticket booking system. But our smart booking system will use our smart NID card .so, Bangladeshi people don't need to buy extra smart card. So anybody can buy ticket just going to the smart booth.

There are many limitation of manual ticket buying system this are.

Limitation of manual ticket buying system:

- People have stand in a long line for a long time.
- For every line there need to be a ticket clerk and a person to maintain the line.
- Because of standing in the line people waste times.
- Even some times people quarrel with each other in the line.
- Lack of security
- High cost

Limitation of other smart ticket buying system:

- Need to buy special Smart card.
- Not enough user friendly.
- Not properly implemented.

- As people have to buy special smart card that’s why people don’t wants the extra work and because of that the smart booth are not used that often.

Advantage of our system:

- The most user friendly system it can be.
- Low cost and high performants system.
- Easy to used.
- Saves time.
- No extra smart card needed

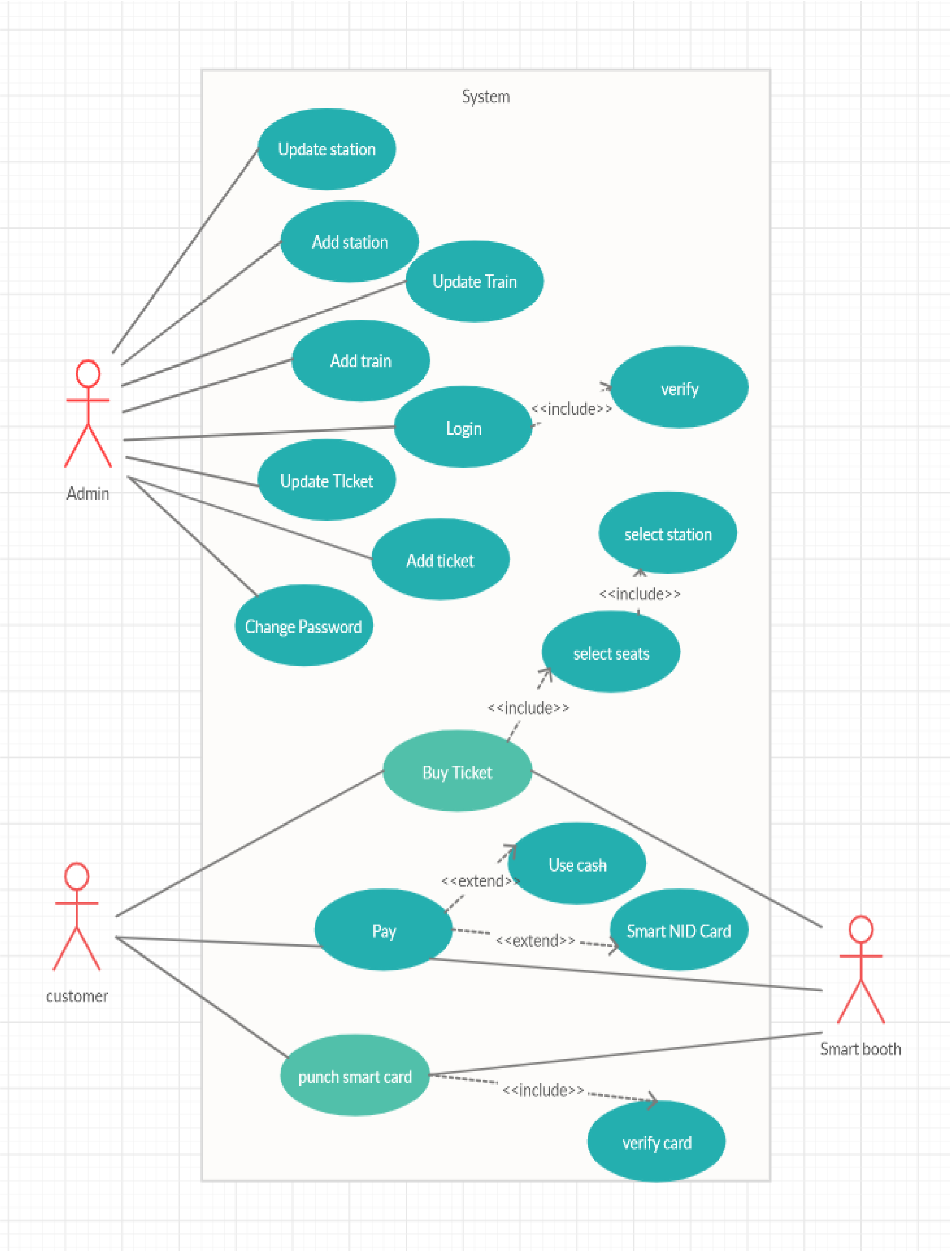
Scope and objectives: Our main objective give user the user friendly and faster system with smooth and secure transaction of money. In our country many people are uneducated. It’s a big disadvantage for digitalization for ticket buying system. So, it’s a big challenge for the developers who are going to develop the smart ticket buying system. The system user interface as friendly as ad uneducated person can operate the booth. In booths there will be a display to show the destination picture will be shown and there will also be a speaker to speak everything we do in the booth. S0, our system will be the best possible user friendly as it can be. It also need to be fast because there will be too much people who will want to buy tickets from the booths. It’s also need to be smooth and bug free because there will be transaction of money. If there is any bugs in the system money transaction may stop or may transferred to another account this is a big loss both customer and the Railway system.

Objectives:

- To provide the user the best user friendly experience.
- Provide ticket buying system without any time wasting.
- Secure money transaction of the customer.
- Provide a system that can be used by primary level educated people too.
- Need to be robast.so no one can hack. As a government project many people will try to hack the system.

Overview of the system: Here is the use case diagram of the system.

Use case diagram:



Use case diagram not for capturing all the details of the system. Only main work.Use case diagram for “what” the systm is not “how” the system work. Here are the ditails of the use case diagram we have drawn.

1.Login: This use case describe how user will login to the system.

Actors: Admin
Pre-condition:Admin already registered in system
Post-condition:user verification starts.

2. Verify: This use case describe how user login will be verify.

Actors: Admin
Pre-condition:Admin need to click in login with password and id.
Post-condition:If the use case varification is successful the user will be logged in.

2. Add station: this use case describe that station can be added to the system.

Actor:admin

Pre-condition:user need to logged in.

Post-condition:If the use case is successful user can able to add station.

3. Add train: this use case describe that train can be added to the system.

Actor:admin.

Pre-condition:user need to logged in.

Post-condition:If the use case is successful user can able to add train.

4. Add ticket: this use case describe that ticket can be added the the system.

Actor:admin

Pre-condition:user need to logged in.

Post-condition:If the use case is successful user can able to add ticket.

5. Update station: this use case describe that station can be Updated to the system.

Actor:admin

Pre-condition:user need to logged in.

Post-condition:If the use case is successful user can able to Update station.

6. Update train: this use case describe that train can be Update to the system.

Actor:admin.

Pre-condition:user need to logged in.

Post-condition:If the use case is successful user can able to add train.

7. Update ticket: this use case describe that ticket can be Updated to the system.

Actor:admin

Pre-condition:user need to logged in.

Post-condition:If the use case is successful user can able to Update ticket.

8. Punch smart card : this use case describe that how customer start ticket buying process.

Actor:customer

Pre-condition:user need to come to the booth.

Post-condition:If the use case is successful card verification will start.

8. verify card : this use case describe that how customer card verify.

Actor:customer

Pre-condition:user need to punch smart card.

Post-condition:If the use case is successful user can buy ticket.

9. Buy ticket : this use case describe that how customer can buy ticket.

Actor:customer

Pre-condition:need sucessful verify card use case.

Post-condition:If the use case is successful user can select seat.

9. Select seat: this use case describe that how customer can select seat.

Actor:customer

Pre-condition:need sucessful buy ticket.

Post-condition:If the use case is successful user can select station.

9. Select station: this use case describe that how customer can select station.

Actor:customer

Pre-condition:need sucessful select seat.

Post-condition:If the use case is successful pay use case start.

10. Pay : this use case describe that how customer can pay ticket bill.

Actor:customer

Pre-condition:need sucessful verify card use case.

Post-condition:If the use case is successful user can select payment method.

10. Smart NID card : this use case describe that how customer can pay bill buy NID card.

Actor:customer

Pre-condition:need sucessful pay use case done.

Post-condition:If the use case is successful customer will get the tickets.

10. Cash: this use case describe that how customer can pay bill buy Cash.

Actor:customer

Pre-condition:need sucessful pay use case done.

Post-condition:If the use case is successful customer will get the tickets .

Selection of model: In this project the requirements are very well known. And as a metro railway ticket management system it will not change in future. Also there no need of partitioning of the project into distinct stages .So “water fall model “is the best choice for implementation of “Dhaka Digital Metro Railway Ticket System”.

The Justification of why we selected water fall model for our project given bellow:

- The water fall model used when requirements are very well known and in this project requirements are well known.
- Technology is properly understood no new technology or unknown technology is implemented.
- Project definition is stable the requirements will not changes.
- Even there are no ambiguous requirements.

- The project is simple so water fall is good for simple and short project.

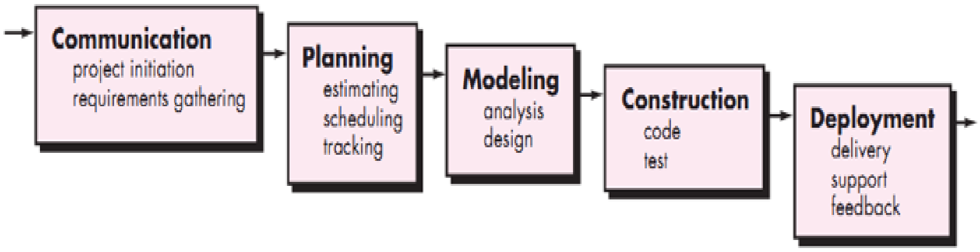


Fig: Waterfall Model

<u>Team structure for the project:</u>

<u>Project manager:</u>

- Manage the whole project.
- Lead the project.
- Develop maintaining plan.
- Control project planning.

<u>Project sponsor:</u>

- Resolve strategic and policy issues.
- Drive the organization.
- Prioritizing the project goal.
- Raise awareness at senior level.

<u>Project Team Members:</u>

- Does the work of documentation and analysing.
- Work with users to ensure the project meet the user need.
- Identify user need.
- Train the user with project.

Project Contributor:

- A project team contributor is a person or an organization.
- Contributors help improve the project through giving valued suggestions.
- Often project team contributors have an interest or concern in the project, so they facilitate successful completion.

System Administrator:

- Management and support.
- Database management and support.
- Backup and recovery works.
- Development and operation of technical testing programmes.

Systems Developers:

- Work with project manager.
- Mainly develop the system.
- Interfaces with other systems.
- Setup and maintenance work.

System tester:

- Main do the work of testing.
- Report the bugs to the developers.
- They make the testing documentation which are going to be tested.

Team leader:

- Every team has a team leader.
- Provide guidance to the team.
- The person who lead a team.
- Takes responsibility for the result of the team work.

Resource requirements: for every software to run there are some hardware and software requirements as well as some human requirements are also needed.

Software requirements:

- Operating System: Windows 10 Professional (64 bit)
- Development Language : php , javascript
- Database: MySQL
- Validation : javascript,php
- Testing: using Xampp local server
- Design:Bootstrap4 ,CSS3 and HTML5

Hardware requirements:

- Processor: Intel-Pentium G4560
- RAM : 500MB
- Hard-Disk space: 10GB
- Display: DB10E-T

Human requirements:

- 4 front-end developer.
- 4 back-end developer.
- 2 system administrator.
- 1 project manager.
- 5 project team members.
- 4 system tester

Functional requirements: In the department of software engineering, a functional requirement defines a function of a software, system or component. Functional requirements can be calculations, data manipulation and processing and other specific functionality .Generally, requirement analyst generates functional requirements after building the use case. But there are many exception because software development process is an iterative process.

Customer NID Validation:

- Check NID card: this function check the NID card is placed on the scanner or not.
- NID card verification: NID card and pin is verified by matching with the database.

Ticket buying:

- Display time table: Display the starting and ending time
- Display station: display starting station and ending station
- Display available seat: to display the seat available seat in each train in schedule.
- Calculate cost: calculate the total cost.
- Pin required: pin is required to complete the process

Add train: admin can add train.

Add station: to add stations for the train to come and go.

Add tickets: tickets can be added.

Update train: train details can be edited and deleted

Update station: station details can be edited and deleted

Update ticket: ticket details can be edited and deleted

Non-functional requirements: In software engineering there are some requirements are not given by the customer but need to be full fill those are non-functional requirement. Non-functional requirements are called quality of a system. Non-functional requirements of the project:

- Performance: its need to be fast enough to give output in seconds.
- Reliability: as there will be money transaction it need to be high reliable and bug free.

- Cost: as a government project in need to be cost effective.
- Time: the project must be completed within due time.
- Security: multiple time wrong pin will suspend the NID card for 1hour.
- Usability: The user experience will the as friendly as it can be.

Constraints limitation for the system: for every project developer develop there is some limitation of that project that can't be done at the first time. And our project meet the following limitations:

1. Single Person can't buy more than 4 tickets at a time.
2. Single person can't buy ticket for multiple stations at a time.
3. Software does not advance ticket buying system more than 10 days
4. The software does not give any concession in fare rates for children, aged people. It give same fare for all people.
5. Ticket price are given manually not automatically generated based on distance.

Conclusion: By developing such a project it was very enriching and informative experience for me. I get to design the project system. I also developed coding skills. By doing many error and solving in the project we understand the importance of planning and designing. The project enhanced my practical knowledge and how to work in pressure. By developing this project it helped me to get some experience of real time application.