****

American International University-Bangladesh (AIUB)

**Department of Computer Science**

**Faculty of Science &Technology (FST)**

**Summer 2020-2021**

**Software Development Project Management**

**Section: A**

**Project Title: Developing a Software Development Project Management Plan for Dhaka Subway Systems Automated Ticket Issuing System**

**Submitted By**

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial No.** | **Student Name** | **ID** | **Program** |
| **1** | **Samima Sultana** | **16-32012-2** | **CSSE** |
| **2** | **Afif Jawed Rafi** | **17-34414-1** | **CSE** |
| **3** | **Ashrafur Rahaman** | **17-35211-2** | **SE** |
| **4** | **Riazul Haque Rana** | **18-36306-1** | **CSE** |

**Index**

**Table of Contents**

1. Title page**…………………………………………………………………………..01**
2. Index**……………………………………………………………………………….02**
3. Revision history page**……………………………………………………………...03**
4. Introduction**………………………………………………………………………..04**
5. Process Model**…………………………………………………………………......04**

5.1 Choose a model**………………………………………………………..04**

5.2 Justify why you chose this model**………………………………..........04**

5.3 Software life cycle flow chart**……………………………………........05**

1. Quality gate for each phase of software development**…………………….............05**
2. List of tasks (work breakdown structure)**……………………………………........05-06**
3. Estimation for each task**……………………………………………………….......06**
4. Schedule the tasks**……………………………………………………………........7-13**
5. List of Milestones**………………………………………………………………….13**
6. Staffing Plan**……………………………………………………………………….14**
7. Monitoring and Controlling Mechanism**………………………………………......15**
8. Risk Management**……………………………………………………………..…...16**
9. List of Deliverables**……………………………………………………………......17**
10. Defect Tracking Process**………………………………………………………......18**
11. Matrices**……………………………………………………………………………18**
12. Post Mortem**……………………………………………………………………….18**

# **3. Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Updated by** | **Update Comments** |
| **1** | 10.07.2021 | Ashrafur Rahaman | 1st Draft |
| **2** | 14.07.2021 | Samima Sultana | 2nd Draft |
| **3** | 20.07.2021 | Afif Jawad Rafi | 3rd Draft |
| **4** | 25.07.2021 | Riazul Haque Rana | 4th Draft |
| **5** | 30.07.2021 | Afif Jawad Rafi | 5th Draft |
| **6** | 01.08.2021 | Riazul Haque Rana | 6th Draft |
| **7** | 03.08.2021 | Samima Sultana | 7th Draft |
| **8** | 04.08.2021 | Ashrafur Rahaman | Final Draft |

**4. Introduction**

An automatic ticket giving structure sells metro tickets. Passengers first input the information into the display and confirm purchase. Then the billing can be done by card, cash or maybe a metro authorized card. After successful transaction the system will dispatch the ticket automatically.

**Requirements:**

The software has the following set of requirements:

· The software will support interface to touch screen monitors as well as keyboard interface.

· The software will support display of the list of incoming trains, their destinations and arrival and departure times, fare, expected travel time

· The software will support multiple ticket purchases simultaneously.

· The software will support limiting the number of tickets purchased at the same time.

· The software will support purchased ticket cancellation by the administrator.

· The software will support credit card transactions and validation.

· The software will support transaction using bill (taka) /coin

· The software will support next and previous navigation during the ticket purchase process.

· The software will support ticket availability information.

· The software will support information display via the web.

· The software will use Oracle database server.

**5.** **Process model**

**5.1** There are many development process models in SDLC. We have chosen an Agile model framework called scrum to build the software.

**5.2** We have chosen an agile model to build the software. Because Agile methods are dynamic. If we get any further changes to the project we can do that without any hassle. Backtracking and feedback are the main advantages of agile development.  
On the other hand, the waterfall model is chosen only if the requirements are fixed and there is no chance of changing the requirements. There are many sectors where changes might be needed in the near future. If we don’t use agile then it would be a big trouble.  
  
That’s why we have chosen Scrum method which is the most popular framework of Agile.

**5.3** **Agile Model life cycle flow chart**



**6. Quality Gets for Each Phase of SW Development**

In agile improving productivity is a prime object but this term has another meaning that commonly associate with custom application. Because it is even more intuitive the application meets specifications and the prevention of defects after the delivery of the application are also common in manufacturing. About quality it is defined as how the delivered project satisfies the customer. There is maintenance which is also considered here. If everything is good then we can say that a good quality project has been delivered.

**7.** **Work Breakdown Structure (WBS)**

**Work Modules for Mobile (Touch) and Web (Keyboard) Platform:**

* 24/7 service

* Ticket availability information display
* Train arrival and departure time display
* Touch screen menu selection
* Source and destination selection
* Multiple ticket issue in one transaction
* Limit the number of ticket issue at the same time
* Cancellation of transactions any time during transaction
* Credit Card transaction
* Coin/Taka recognition and acceptance

**8.** **Estimation for Each Task**

The effort estimation applies to individually all those modules for **“Web Application”** development as followed by:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Project Manager and experts** | **Front-end developer** | **Back-end developer** | **QA** |
| Conception/design | 12% | 5% | 7% | 3% |
| Development | 5% | 14% | 25% | 15% |
| UAT | 3% | 3% | 6% | 5% |
| Support | 3% | 3% | 5% | 6% |
| **Total** | **23%** | **25%** | **43%** | **29%** |

**SDLC Cost Estimation (Web/Keyboard Input Part):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective** | **Working Days** | **Working Hours** | **Cost/HR (BDT)** | **Total Cost (BDT)** |
| Gathering Information | 3 | 24 | 1000/- | 24000/- |
| Planning | 2 | 16 | 1000/- | 24000/- |
| Design | 7 | 56 | 1500/- | 72000/- |
| Content Writing and Assembly | 5 | 40 | 1500/- | 72000/- |
| Coding | 13 | 104 | 2500/- | 240000/- |
| Testing, Review and Launch | 6 | 48 | 1500/- | 72000/- |
| Total | 36 | 288 |  | 504000/- |

**Full web part working time:**

· 3 Back End developer working full time for 36 days

. 1 Front End developer working full time for 36 days

. 2 API developer working full time for 36 days

The effort estimation applies to individually all those modules for **“Mobile Application”** development as followed by:

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Team Leader** | **UI/UX developer** | **Back-end developer** |
| Conception/design | 14% | 8% | 16% |
| Development | 9% | 6% | 14% |
| UAT | 6% | 5% | 8% |
| Support | 6% | 5% | 7% |
| **Total** | **35%** | **24%** | **45%** |

**SDLC Cost Estimation (Mobile/Touch Screen Part):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective** | **Working Days** | **Working Hours** | **Cost/HR (BDT)** | **Total Cost (BDT)** |
| Strategy | 3 | 24 | 500 | 12000 |
| Designing/Analysis | 5 | 40 | 1500 | 60000 |
| UI/UX Design | 5 | 40 | 1300 | 52000 |
| Coding/Development | 22 | 176 | 2000 | 352000 |
| Deployment | 5 | 40 | 600 | 24000 |
| **Total** | **40** | **320** |  | **500,000/-** |

**Full mobile part working time:**

* 4 Back End developer working full time for 40 days
* 2 UI/UX developer working full time for 40 days
* 1 Team leader working full time for 40 days

# **Cost of System Interconnection:**

After calculating the module costing we need to calculate ERP system cost. Interconnecting the modules will require the development of an API to make them communicate. Developing a similar API for each module to enable communication between the modules is cost-effective.

**There are four steps needed to develop an API:**

* Research to understand the data structures and the communication protocol between each module.
* Build an API prototype
* Develop the API
* Document the API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective** | **Working Days** | **Working Hours** | **Cost/HR (BDT)** | **Total Cost (BDT)** |
| Developing Prototype | 15 | 120 | 300/- | 36000/- |
| Build & Test API | 30 | 240 | 700/- | 168000/- |
| API Documentation | 5 | 40 | 300/- | 12000/- |
| Total | 50 | 400 |  | 216,000/- |

**Full API part working time:**

* 2 API developer working full time for 50 days

# **Cost of Data migration and validation:**

Migrating data from the old system to the new one will require data extraction, by extraction it means data fetching. Data migration will be applied here. After that it will be validated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective** | **Working Days** | **Working Hours** | **Cost/HR (BDT)** | **Total Cost (BDT)** |
| Data Extraction | 20 | 160 | 600/- | 96000/- |
| Data Migration | 20 | 160 | 700/- | 112000/- |
| Data Validation | 10 | 80 | 700/- | 56000/- |
| **Total** | **50** | **400** |  | **264,000/-** |

**Full Data Migration and Validation part working time:**

* 2 database administrator working full time for 50 day
* 1 ETL experts working full time for 50 days

# **Test cost estimation:**

To thoroughly test software of this level of complexity, it will require:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective** | **Working Days** | **Working Hours** | **Cost/HR (BDT)** | **Total Cost (BDT)** |
| Quality Testing | 30 | 240 | 1000/- | 240000/- |
| **Total** | **30** | **240** |  | **240000**/- |

**Full QA part working time:**

* 3 QA Engineers working full time for 30 days

**9. Task Scheduling  
  
 Web/Keyboard Input Part:**

**Schedule:**

|  |
| --- |
| **Web/Keyboard input Part** |
| Gathering Information 70% 12/7/2021-16/7/2021 |
| Planning 85% 17/7/2021-20/7/2021 |
| Design 0% 18/7/2021-22/7/2021 |
| Content Writing and Assembly 0% 23/7/2021-26/7/2021 |
| Coding 0% 27/7/2021-30/7/2021 |
| Testing ,review and Launch 0% 31/7/2021-2/7/2021 |

**Gantt chart:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12-JULY-2021 | | | 15-JULY-2021 | | | | 18-JULY-2021 | | | | | 23-JULY-2021 | | | | 27-JULY-2021 | | | | 31-JULY-2021 | | | |
| 12 | 13 | 14 | | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | | 31 | 1 | 2 |
| M | T | W | | T | F | S | S | M | T | W | T | F | S | S | M | T | W | T | F | | S | S | M |
|  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |

**Mobile/Touch Screen Part:**

**Schedule:**

|  |
| --- |
| **Mobile/Touchscreen Part** |
| Strategy 80% 12/7/2021-16/7/2021 |
| Design 45% 17/7/2021-20/7/2021 |
| UI/Ux Design 0% 21/7/2021-23/7/2021 |
| Coding 0% 24/7/2021-28/7/2021 |
| Deployment 0% 29/7/2021-2/7/2021 |

**Gantt chart:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12-JULY-2021 | | | | | 17-JULY-2021 | | | | 21-JULY-2021 | | | 24-JULY-2021 | | | | | 29-JULY-2021 | | | | | |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| M | T | W | T | F | S | S | M | T | W | T | F | S | S | M | T | W | T | F | S | S | M |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**10. List of Milestones**

**Web/Keyboard Input Part:**

|  |  |
| --- | --- |
| **Milestones** | **Date** |
| Conception/Design Documentation | July 11 – july16 |
| Implementation | July17- july23 |
| Demo Presentation | July - 24 |
| Final Launch | August-1 |

**Mobile/Touch Screen Part:**

|  |  |
| --- | --- |
| **Milestones** | **Date** |
| Analysis and Design Documentation | July 11 -July 18 |
| UI/UX Design | July 19 - July 23 |
| Implementation | July24 – July28 |
| Testing and QA | July 29 – July31 |
| First Deployment | August 1 |
| Feedback analysis and Next Increments | August 2- August 4 |

**11.** **Staffing Plan**

**Developer Team (Web):**

|  |  |
| --- | --- |
| **Designation** | **Number** |
| Back End Developer | 2 |
| Front End Developer | 1 |

**Developer Team (Mobile):**

|  |  |
| --- | --- |
| **Designation** | **Number** |
| Back End developer | 3 |
| UI/UX Developer | 2 |
| Team Leader | 1 |

**Other Staff:**

|  |  |
| --- | --- |
| **Designation** | **Number** |
| System Administrator | 1 |
| QA Engineer | 2 |
| ETL Expert | 1 |
| API Developer | 2 |
| DB Administrator | 2 |

**12. Monitoring and Controlling Mechanism**

**Changes:**

* Git is used as a version control system.
* Only the Team Leader and Project Manager have authority to give access to change.

**Meetings:**

* Project manager will supervise project Kickoff, status, stakeholder and review.
* Team leader will organize change control meetings.
* Daily Scrum meetings are to be held under Team Leader or Scrum master.
* Scrum master and the team of developers will call the weekly meeting with product owner.

**Reports:**

* Monthly reports will be shown to The Directors.
* Project Manager have access to all the weekly, monthly and change control report
* Team members provides a short document to the Team Leader to request access to change control

**Maintaining Schedule:**

* Employees are provided proper flexibility but are given time boxes for each task. This will increase the efficiency of the employees.
* Meeting relevant employees and managers are must to attend and results to be reported on schedule.

**13.** **Risk Management**

For managing uncertain issues during the development process, risk management is implemented. Some risks that could happen during doing this project and also the Mitigation plan for are given below:

|  |  |
| --- | --- |
| **Risk** | **Mitigation plan** |
| Developing the wrong user interface | Prototyping, task analysis, user involvement |
| Real time performance problems | Simulation, prototyping, technical analysis |
| Late changes to requirement | incremental change |
| Unrealistic time and cost estimation | Multiple estimation technique, incremental development, |

**14. List of Deliverables**

The framework documents consist of the operating concepts. This consists of a device introduction, a review of the operating system and the satisfactory passage of the approval test.

**Deliverable work Products:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Standard** | **Preparer** | **Reviewer** | **Date** |
| **Software Design Specification (SDS)** | IEEE | IEEE | Project  Manager | July 12 |
| **End-user Documentation** | IEEE | Technical  Writer | Project  Manager | August 3 |
| **Software** | IEEE | Requirements | Requirements | July 12 |
| **Software Project Management Plan (SPMP)** | IEEE | Project  Manager | Project Manager | July 18 |
| **Software Test Plan (STP)** | IEEE | Verification  Engineer | Project Manager | July 13 |
| **Software Quality Assurance Plan (SQAP)** | IEEE | Quality  Analyst | Project Manager | July 15 |
| **Software Verification** | IEEE | Verification  Engineer | Project Manager | July 20 |

**15. Defect Tracking Process**

Monitoring the solved defects record is very important. It will test engineers to recognize the past defects. This helps a test cycle to continue one after another up to the end of a final product.

**Ways of detecting the defect or bug are given below:**

* Assigned specific developers to monitor and repair defects in the design and test phase using diagnostic methods so that both positive and negative approaches will interpret the test scenarios.
* Implementing defect monitoring tools.

**16. Matrices**

* **Schedule:** It refers to project milestone.
* **Lead Time:** Thetime difference of delivering a product and the actual delivery.
* **Testing Capability:** The unit evaluations were over time activated test Integration Quantity.
* **Coding Coverage:** How much code is covered
* **Defect tracking ability:** Number of defects in code and number of software code defects.

**17. Postmortem**

This project was launched to address some of the problems faced by the passengers during a ticket booking period like passengers face many problems, especially when they buy tickets physically. Three prototypes have to be delivered : a graphical interface, functional prototype, a system integration prototype. Analysis is started before project planning is finished. We will be to completed this project successfully without any major interruption.