Implementation Phase

**Loza Tadesse, Amir Kemal,****Nahome Hailemichael,****Yanet Woldemichael**

Contents

[Synopsis 2](#_Toc90257741)

[Organization document structure 4](#_Toc90257742)

[Test Plan 5](#_Toc90257743)

[Conversion Strategy 7](#_Toc90257744)

[Post-Implementation activities 8](#_Toc90257745)

## Synopsis

Planning:

In the planning phase, we layout the description, needs and requirements for our system analyst. We figure out a methodology for our system and decide to use waterfall methodology for the suitable of the project are it uses clear structures with defined set steps requirement gathering and documentation, System design, Implementation, Testing, Delivery/deployment, and Maintenance. The financial spreadsheet shows our ROI coming out to be 2% with a breakeven point of 0.53. Finally, our work breakdown structure details the amount of work needed to get done from start to end of the project.

Analysis:

In the analysis phase, we determine the overall requirements of our system, and review the previous system documents to determine what is needed. Our functional and non-functional requirements are then laid out in detail. We then build our use-case diagrams that lay out the simple steps of an applicant sending out an application to the server and admissions, followed by a use case description for those use cases. An activity diagram is then built to show the flow of how both subscribers and non-subscribers can get information from the schedule. A class diagram is made to show the details and needs of the objects, methods, and operations we needed to include in our system we also included e multiplicity for all non-generalization associations in the diagram.

Design:

In the design phase, we come up with a design strategy on how to properly design the system, in which we decide upon a packaged software to complete the design after evaluating them on an alternative matrix and evaluating packaged software to be a higher score than outsourcing and in-house development. Our method specification forms layout the methods, triggers, and arguments for our classes. Our relational database relates our class and actors together in terms of how many there are of each and how those would link up. UI mockups are designed to show off what our system’s UI would look like when completed. Finally, our physical architecture lists our hardware, software and network specifications followed by presenting those requirements on a deployment diagram.

Implementation:

In the implementation phase we start off by determining the organization structure of our documentation, explaining the contents of our system and the various commands and specifications that go with the system, followed by various how-to manuals on how to run and operate the system. The test plan is then written up to show the process of testing our system. With the test plan in mind, we work on the conversion strategy, in which our system is using the Parallel Conversion Strategy to implement the system alongside with the old system to get previous workers to slowly get used to the new system. Finally, a plan is developed for post implementation of our system, detailing how our system will function after its implementation, and how to keep it stable through the coming years.

## Organization document structure

Diagram

Description automatically generated

## Test Plan

For the test plan of our system, we will use 4 stages of testing namely Unit testing, Integration testing, System testing, and Acceptance testing. First, we will be using unit testing to test each class in our system and using the black-box test plan source. For the User class we will check if the attributes username, id, address and the request Service method works properly. In this case, the test plan is developed directly from the specification of our class: each item in the specification becomes a test, and several test cases are developed for it. For the schedule class we will test that the date, time attributes and the calculateEstimate method are accurate. Going next to the driver class, the location should be tested.

Second, we plan on using integration testing to check if the classes are interacting properly. The integration tests we will be doing for the User interface testing for the user classes if the ReserveVan, TrackVan, ViewRoute Methods link with the Van class without error. Integration tests for the Driver class include testing if the methods UpdatdeSchedule interact with the schedule class and UpdateSeatAvailabilty interact with the Van class appropriately.

Then, System testing will be the third stage for our test plan. System tests examine how well the system meets both the functional and nonfunctional requirements. We will test that by requirements testing if the users in our system are able to submit information, view schedule, reserve van, track van and view seat availability. Then we will test if our system operates in IOS, Android, and as a webapp, test that it connects to the internet, test if it is able to access DSU’s cloud server, test if the system can update the van’s location every 3 second, and test if only authenticated users can use the system.

The final stage of testing for our system is acceptance testing. We will need to use alpha testing to check if the users are comfortable with using a public transportation van in Madison. Then use beta testing to track if the system is solving the transportation problem in Madison and attracting students and faculty to DSU.

## Conversion Strategy

Conversion style

Parallel

Conversion location

Pilot conversion

Conversion modules

Modular conversion

## Post-Implementation activities

The goal of post-implementation activities is the institutionalization of the use of the new system—i.e., to make it the normal, accepted, routine way of performing the business processes. Post-implementation activities attempt to refreeze the organization after the successful transition to the new system.

**Training**

We have decided to use Computer based training (CBT) in which the training program is delivered via computer or over the Web. Our computer-based training programs will comprise of recorded videos of people using our system, giving comments. Animation and audio version will be provided accordingly to the preference of our audience. Computer based trainings is typically more costly to develop but is cheaper to deliver because no instructor is needed to provide the training. And since our target trainees are college students, they will not have problem accessing our resources as they equipped with the proper technological tools. CBT has the greatest reach—the ability to train the most users over the widest distance in the shortest time—because it is much simpler to distribute than classroom and one-on-one training, simply because no instructors are needed.

**System Support**

Providing system support means helping the users to use the system. Usually, this means providing answers to questions and helping users understand how to perform a certain function; this type of support can be thought of as on-demand training.

We plan to use online support because it is the most common form of on-demand training. It will include the documentation and help screens built into the system, as well as separate websites that provide answers to frequently asked questions (FAQs), which enable users to find answers without contacting a person. We have adopted similar feature from other transport services. Since our service is only going to implemented on a short scale it will not need massive customer support.

**System Maintenance**

We plan to maintain our system and make sure it continues to meet business needs by assigning an active and participating project manager and whenever a potential change to the system is identified a change request will be prepared and forwarded to our project manager. We will utilize common source of change requests which is enhancement to the system from users. As users work with the system, they often identify minor changes in the design that can make the system easier to use or identify additional functions that are needed. We will have close contact with our employees and have consistent line of communication.

**Project Assessment**

The goal of project assessment is to understand what was successful about the system and the project activities (and, therefore, should be continued in the next system or project) and what needs to be improved. We carry out meetings with the project manager once a week and discuss about achievements and things we need to work on. Each team member will write reports or evaluate themselves and present it to the project manager after which he will assess it and point out things that need to be worked on and appreciate those who performed well. The system is then reviewed to understand the extent to which the proposed costs and benefits from the new system identified during feasibility analysis were recognized from the implemented system. After the project is up and running all team member including the project manager will reexamine the project as a whole and decides if it meets the anticipated benefit of the project.