

|                           |                  |
|---------------------------|------------------|
| MYES Bus Ticketing System |                  |
| Project Plan              | Date: 10/04/2021 |

# MYES Bus Ticketing System Project Plan



## Revision History

| Date       | Author(s)    | Description  | Version |
|------------|--------------|--|---------|
| 10/04/2021 | Serdar Mumcu | Initial Version  | v1.0    |
| 17/04/2021 | Serdar Mumcu | Iteration1 review's accepted changes have been fixed in the document.<br><br>Risks are planned to be handled in Iterations. (and in other related documents as well)<br><br>development environment, build and test tools are specified. | v1.1    |
| 30/04/2021 | Serdar Mumcu | Iteration3 plan and lessons learned section are updated  | v1.2    |
| 08/05/2021 | Serdar Mumcu | Lessons learned is updated   | v1.3    |
| 23/05/2021 | Serdar Mumcu | Iteration4 objectives and  | v1.4    |

|                           |  |                                      |  |
|---------------------------|--|--------------------------------------|--|
| MYES Bus Ticketing System |  |                                      |  |
| Project Plan              |  | Date: 10/04/2021                     |  |
|                           |  | target velocity sections are updated |  |

## 1 Introduction

MYES Bus Ticketing system is planned as being an online multi-tenant SAAS solution for the passengers to search for a trip, to make/cancel reservations and to buy a ticket online without having to visit the bus company's office locations in person.

In this Project Plan, we will be covering how the project is organized, what the practices and measurements are and what the high-level objectives and milestones are as well as production deployment strategy of the MYES Bus Ticketing project.

## 2 Project organization

In this project, team members and their roles are as follows:

- Mustafa Ilkkan: Software Architect and Software Developer
- Yiğit Erkal: Tester and Software Developer
- Eren Erol: System Analyst and Software Developer
- Serdar Mumcu: Project Manager and Software Developer

The Project Manager (Serdar Mumcu) is responsible for planning and managing project objectives and iterations, coordinating the communication between stakeholders, ensuring the project will meet the objectives defined by tracking the progress. He is also responsible for preparing a coarse-grained project plan, preparing fine-grained iteration plans, making risk management and definition work items in the project.

The Software Architect (Mustafa Ilkkan) is responsible for giving the key technical decisions that affect the overall design and implementation of the project. He is also responsible for creating and keeping the Architectural Notebook up-to-date.

The System Analyst (Eren Erol) is responsible for understanding the problem domain by communicating with stakeholders and identifying the business requirements accordingly. He is also responsible for creating and revising the vision document, maintaining the glossary, establishing system-wide requirements, making use-case modeling and defining use-cases.

The Software Developers (Mustafa Ilkkan, Yiğit Erkal, Eren Erol, Serdar Mumcu) are responsible for creating a design to solve the problems specified in the requirements that is aligned with the Architecture, performing implementation activities, writing unit-tests and making necessary integrations.

Tester (Yiğit Erkal) is responsible for creating test cases, performing the necessary tests to conduct verification and validation on the product and maintaining a test log.

A regular weekly meeting will be organized for communication among team members regarding evaluation of the project progress and making critical decisions together. Since we have a remote team, meetings will be held online on zoom or other online collaboration platforms which will boost team collaboration and utilize team performance.

## 3 Project practices and measurements

As the management and technical practice for this project, the Open Unified Process Model is selected. The process model is an iterative and incremental approach that has a structural lifecycle. It is a use-case driven, architecture centric and risk driven process model. This model consists of iterations and these iterations are just like time-boxed mini projects. We have selected this process model since the following practices of the Open UP model will be very important for our needs:

- Tackling high-risk and high business value issues in earliest possible iterations
- Continuously engaging customers for evaluation, feedback and requirement

|                           |                  |
|---------------------------|------------------|
| MYES Bus Ticketing System |                  |
| Project Plan              | Date: 10/04/2021 |

- Building a architecture in early iterations
- Applying use cases

Typically, this process model has 4 phases: Inception, Elaboration, Construction and Transition. However, we will not include the transition phase. We will have 1 iteration for inception, 2 iterations for elaboration and 1 iteration for the construction phase of the process model.

For each iteration, we will be preparing an iteration plan that consists of the objectives and evaluation criteria of the particular iteration so we will be evaluating the success of each iteration at the end of the iterations. In every iteration, we will be creating an iteration plan for the next iteration as well.

We will be using Python 3.x as the programming language and Django 3.x as the web framework. VS Code as our Integrated development environment. We are going to be working on a dockerized development environment. We will be using MySQL or PostgreSQL as our Database Engine. Tests will be conducted manually by our Test specialist.

We will be using [Github](#) as the configuration management tool for the project artifacts and source code. We are planning to use github as a collaboration tool for our development activity as well.

For the measurement of the project, we will be tracking progress by using a Work Items List document. This document will be a live document and we will be updating it throughout the project. We will use the Agile Size Estimation method; we will be giving points to the work items and we will be measuring and evaluating the velocity.

## 4 Project milestones and objectives

| Iteration | Primary objectives (risks and use case scenarios)  | Scheduled start or milestone | Target velocity |
|-----------|--|------------------------------|-----------------|
| I1        | Objectives<br>1. Revising Vision and Glossary Documents<br>2. Revising Current Iteration Plan<br>3. Preparing Next Iteration Plan<br>4. Creating a Project Plan<br>5. Preparing Use Case Specification Document and use-case diagram<br>6. Starting Work Items List Document<br>7. Preparing a Risk List<br>8. Preparing System-wide Req. Spec. Document | 01/04/2021 - 15/04/2021      | 53              |
| I2        | Objectives<br>1. Revising Current Iteration Plan<br>2. Updating the Project Plan<br>3. Writing More Use-cases in fully dressed format<br>4. Preparing Design Document  | 15/04/2021 - 06/05/2021      | 82              |

|                           |                  |
|---------------------------|------------------|
| MYES Bus Ticketing System |                  |
| Project Plan              | Date: 10/04/2021 |

|    |  |                         |    |
|----|--|-------------------------|----|
|    | 5. Starting Architectural Notebook<br>6. Preparing Test Cases<br>7. Handling of Risk #1 and Risk #2<br>8. Implementing UC1, UC3, UC4 and UC5<br>9. Deploying First Running Partial Software to a Cloud Platform<br>10. Testing Partial Software<br>11. Preparing Next Iteration Plan   |                         |    |
| I3 | Objectives<br>1. Revising Current Iteration Plan<br>2. Preparing Next Iteration Plan<br>3. Finalizing the Architecture Document<br>4. Extending Design Document<br>5. Writing remaining use-cases in a fully dressed format<br>6. Writing related test cases<br>7. Handling of Risk #5 and Risk #6<br>8. Implementing UC2, UC8, UC9 and UC11<br>9. Testing Partial Software<br>10. Deploying Second Running Partial Software to the Cloud Platform | 06/05/2021 - 03/06/2021 | 72 |
| I4 | Objectives<br>1. Revising Current Iteration Plan<br>2. Extending the Design Document<br>3. Writing related test cases<br>4. Implementing UC7 and UC12<br>5. Testing First Release<br>6. Handling of Risk #4 and Risk #8<br>7. Deploying First Release to the Cloud Platform<br>8. Finalizing all artifacts   | 03/06/2021 - 24/06/2021 | 61 |

## 5 Deployment

We are planning to create a dockerized development environment. Therefore, partial software that we will be creating will be running in a container environment so that we will be provisioning a virtual machine with docker on a cloud platform, and it will automatically download our final docker image from Docker Hub for our releases.

For that purpose we will be using docker machines to provision a virtual machine in any cloud platform. In the future we will be using Docker Swarm to scale our application with respect to the load we are having. That way,

|                           |                  |
|---------------------------|------------------|
| MYES Bus Ticketing System |                  |
| Project Plan              | Date: 10/04/2021 |

we are planning to minimize our costs. This environment will also support rolling updates that means we will be able to update each instance gradually and we will have a chance to rollback by executing a single command.

## 6 Lessons learned

- Making a regular meeting with team members seems critical to be able to satisfy the requirements.
- In regular meetings, team members should ask about the issues they are having to the other team members so that a solution can be created or an action plan can be decided together.
- In Reviews, we need to specify the details in a more clear manner.
- We need to communicate more frequently to monitor the progress and take the necessary actions on time.
- Team members should be aware of the changes that may have effect on other artifacts that are primarily managed by the other team members.
- We had better reviewed all the provided artifacts and should not omit a document review because it has already been reviewed previously by other teams. New additions to the software may have effects on every artifact so all of the artifacts should be reviewed.
- If any artifact is missing or not provided, we need to communicate with team members and have them complete missing artifacts.
- Test logs should be created as separate documents to be able to distinguish each test run from each other.
- In System-Wide Requirements and Architecture Notebook documents, all the requirements should be numbered so that we can refer to them from other documents easily.