Software Project Management Plan

Document owner: Raul Farkas is responsible for developing and maintaining this document.
Date: January 05, 2020
Version: 1.0

Revision History

Version	Date	Name	Description
1.0	27/12/19	Raul Farkas	Initial document

1. Overview

1.1 Purpose and scope

Team 3 is interested in updating MarconiTT, a web application used internally at I.T.I. G. Marconi to manage classroom bookings. This update includes three important changes: the addition of new features that facilitate the use of the application, the encapsulation of the web application inside a containerized environment and the displacement of the entire webserver, that contains MarconiTT and other applications, from edu-x04 to edu-x08, a more modern server.

The new features that are to be implemented will allow users to book classrooms for up to 10 hours for multiple days in consecutive weeks using the same form, as opposed to having to book for each day individually. Users will also be able to see previous bookings filtered by day thanks to a small calendar that will be implemented in the "Gestisci prenotazioni" section, as opposed to having all previous bookings in one list with no filters. The MarconiTT logo will be updated with a new one that will be created using the current I.T.I. G. Marconi logo.

The encapsulation of MarconiTT inside a containerized environment eliminates most of the difficulties that developers encounter when updating a software on production servers, because the code will reside inside an identical virtual environment on both the development and production machines.

Moving the webserver and the MarconiTT web application on edu-x08(server) is a fundamental step because the Debian version currently installed on edu-x04 is too old and does not receive security patches any more. This displacement also allows for increased performances as edu-x08 is a virtual machine hosted on a more powerful server compared to the one where edu-x04 currently resides.

1.2 Goals and objectives

The overall objective is to add new features that increase usability and to containerize the entire application so that further developments are easier to implement on the production server.

Project Goals

1. Increase user satisfaction by allowing them to perform basic operations quicker.
2. Decrease the amount of time needed to update the application

3. Increase overall performance

Project Objectives

1. Update the booking form so that it allows users to book the same classroom for multiple hours on multiple days in consecutive weeks.
2. Update the "Gestisci prenotazioni" section by adding a calendar that allows users to filter previous bookings based on the selected date.
3. Create two containers that hold the Database and the Backend application using Docker and Docker-Compose
4. Create a script that can be used to update the timetable on MarconiTT inside the containerized environment
5. Move the database and the backend application inside the Docker containers
6. Move the Webserver with the containerized version MarconiTT on a new server (edu-x08)

1.3 Project Deliverables

Date	Deliverable
04/04/2019	Initial product release with no update functionalities
05/01/2020	Software Project Management Plan, Project Charter, Project Model Canvas, Business Model Canvas, Software Requirements Specification
13/01/2020	Final product release with all functionalities on production server

1.4 Assumptions and Constraints

1.4.1 Assumptions

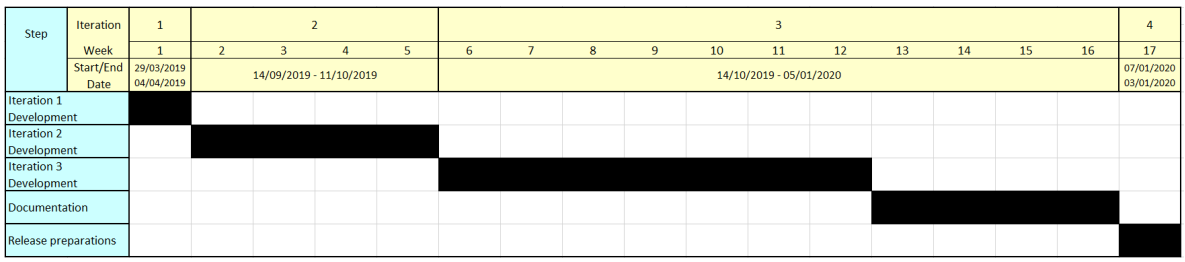
1. A working version of MarconiTT is available and ready to be updated with new features
2. The AD(Active Directory) of the school is working and can be used for authentication on MarconiTT through an API
3. The server that will host the new version of MarconiTT has enough memory, disk-space and supports Docker

1.4.2 Constraints

1. In order to update the timetable on MarconiTT an updated version of the timetable files need to be provided.

1.5 Schedule

1.5.1 Schedule summary



1.6 Success Criteria

The web application MarconiTT needs to be fully functional inside the containerized environment on the server edu-x08. The timetable change process needs to work correctly without errors or data loss.

1.7 Definitions

MarconiTT - the product that is being described here; the application in which team 3 is adding new features and applying changes
Project - activities that will lead to the production of the new version of MarconiTT
Client - the organization for which this application is being built
ITI G. Marconi - a technical institute in Verona, the client of this application
User - the person or the people who will actually interact with MarconiTT
Use Case - describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes
Scenario - one path through a use case
Actor - user or other software system that receives value from a use case
Developer - the person or organization developing the system, also sometimes called the supplier
Stakeholder - anyone with an interest in the project and its outcomes. This includes clients, customers, users, developers, testers, managers and executives
Container - a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another
Booking - a reservation of a classroom or a lab for one or more hours in a day
Timetable - a table that describes classrooms occupations for every hour
Backend - parts of a computer application or a program's code that allow for it to operate and that cannot be accessed by a user
Frontend - all software or hardware that is part of a user interface. Human or digital users interact directly with various aspects of the front end of a program, including user-entered data, buttons, programs, websites and other features
edu-x08 - the production server that runs Debian 11, docker and docker-compose

2. Startup Plan

2.1 Team Organization

Role	Actor(s)	Responsibility
Project Manager	Raul	Call team meetings, break out tasks, assign them to teammates
Developer	Raul, Pietro, Samuele, Mateo	Develop software based on requirement
Tester	Samuele, Mateo	Write test cases, perform testing operations and report issues
Architect	Raul, Pietro, Mateo	Specify internal workings of the software

2.2 Project Communications

Event	Information	Audience	Format	Frequency
Daily Team Meeting	Task status: completed since last meeting & planned for next. Obstacles encountered.	All Team members	Informal meetings	Daily
Project Status Report	Review finished tasks and items, review any problems	All Team members, Stakeholders	Informal meeting	Iteration closeout, as needed (in case of issues)

2.3 Technical Process

An iterative and incremental development process is planned. Feedback will be used from each iteration to improve the next. New Iterations will build upon previous ones and will incorporate more features as time allows.

2.4 Tools

- Programming, Markup Languages, Other Languages... - Python, JavaScript, SQL, HTML, CSS, JSON, YAML
- Operating System - Linux
- Version Control - all work products will be stored in a GIT repository
- Development Tools - Visual Studio Code
- Frameworks and Technologies - Angular JS, Express(Node), Mysql, Docker, Docker-Compose

3. Work Plan

3.1 Release Plan

3.1.1 Plan by Feature

Iteration 1 (29/03/2019 - 04/04/2019)

Summary: Encapsulate MarconiTT inside Docker containers and add new Features

Feature / Deliverables	Estimated Effort	Actual Effort
Learn how to use docker	6	12
Encapsulate MarconiTT inside Docker containers	8	5
Add multiple days booking feature	7	7
Add calendar feature and add new logo	8	8

Iteration 2 (14/09/2019 - 11/10/2019)

Summary: Create Python scripts that allows for automatic timetable update. Move Webserver and MarconiTT on edu-x08.

Feature / Deliverables	Estimated Effort	Actual Effort
Create Python scripts for automatic timetable updates	20	20
Move MarconiTT on edu-x08	8	10
Move webserver with AD access API on edu-x08	3	3

Iteration 3 (14/10/2019 - 05/01/2020)

Summary: Write software documentation. Add new API. Tweaks and bugfixes

Feature / Deliverables	Estimated Effort	Actual Effort
Write software documentation	16	16
Add ETL API	2	2
Tweaks and bugfixes	15	18

Iteration 4 (07/01/2019 - 13/01/2019)

Summary: Update software on production server. Check for minor bugs. Prepare for release

Feature / Deliverables	Estimated Effort	Actual Effort
Update software on production server	5	
Bug fixes / testing	6	
Product release	3	

3.2 Iteration plans

3.2.1 First iteration

The backend and the Database of MarconiTT are contained inside Docker containers and work properly. In case of crashes, the containers restart automatically. Users can now book classrooms for multiple days with just one click. Users can also filter by date previous bookings.

3.2.2 Second iteration

A working copy of MarconiTT and webserver can now be found on production server(edu-x08). Administrators can upload the updated timetables as .txt files inside a specific directory and the Python Updater automatically updates the MarconiTT database.

3.2.3 Third iteration

The developers responsible with processes involving ETL can now use the MarconiTT API to save data related to the timetable and to bookings.

3.2.4 Forth iteration

The software on edu-x08 is updated at its latest version. Production tests are required in order to make sure that the software is ready to be finally released.

4. Control Plan

4.2 Configuration Management Plan

The following procedure is to be used when making changes to all baselined work products:

1. All project work products will be stored in a GIT repository on GitHub. The Database is also under VC on a separate repository.
2. All documentation is also under VC on GitHub (https://github.com/TrackyRaul/2020_5CI_team3_Farkas)
3. The change control procedure once a product is baselined is:
    1. Anyone wanting to make a change has to create a new branch that is based on the "develop" branch on GIT. Once the changes have been made, a pull request can be created to merge from the new feature branch onto the "develop" branch
    2. If the pull request is approved, the new changes will be included in the next production release on the "prod" branch
    3. If the pull request is denied, further discussions are needed in order to figure out how the suggested changes can be adapted or improved.

5. Product Acceptance Plan

At the conclusion of each iteration, the product needs to be tested in an environment similar to the one used on the Production server. Because the application is encapsulated inside Docker containers, testing can be easily done on any machine that supports Docker, as the underlying OS is the same inside the containers.