VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
INSTITUTE OF COMPUTER SCIENCE
INFORMATION TECHNOLOGIES STUDY PROGRAM

Problem-based Project

Tower Defense. 2D game with strategy and AI features.

Group name : **Otojus**
Done by: : **Oleh Petrov,**
Nojus Vislobokovas
Supervisor: : **Linas Būtėnas**

Vilnius

2022

1 Technologies and tools that will be used:

The client-side Windows game is made in Unity Game Engine using the C# programming language. Code will be written in Visual Studio code.

To draw textures we will use the paint tool SAI.

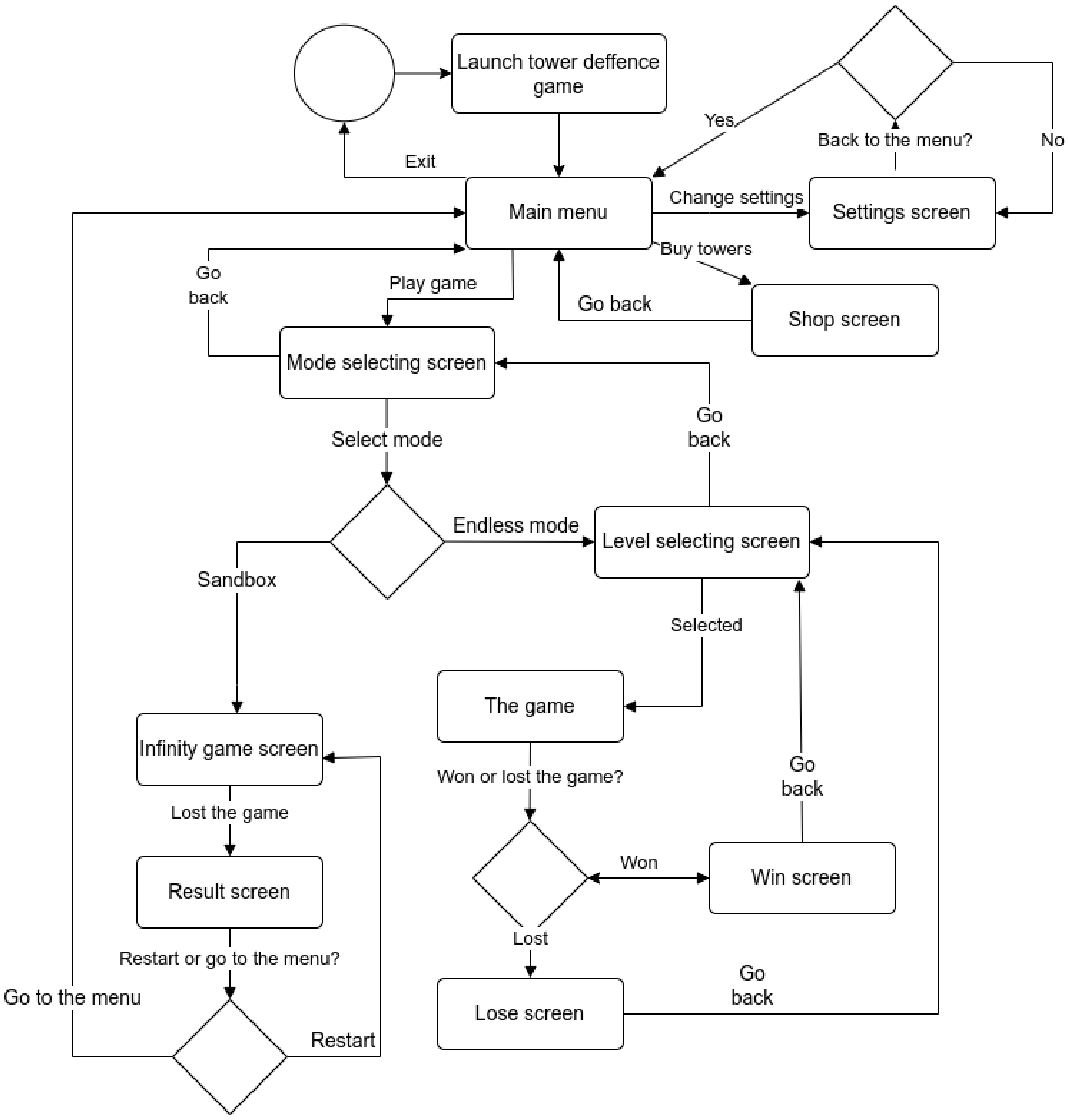
The web server is programmed by C# language. Docker is being used inside of a Linux virtual machine. It is running a REST API, which enables sending and receiving data from the client-side app. It is also responsible for hosting media files. The web server securely communicates with the client-side over HTTPS

For the database part, PostgreSQL relational database management system is used. It is running inside of a Linux virtual machine.

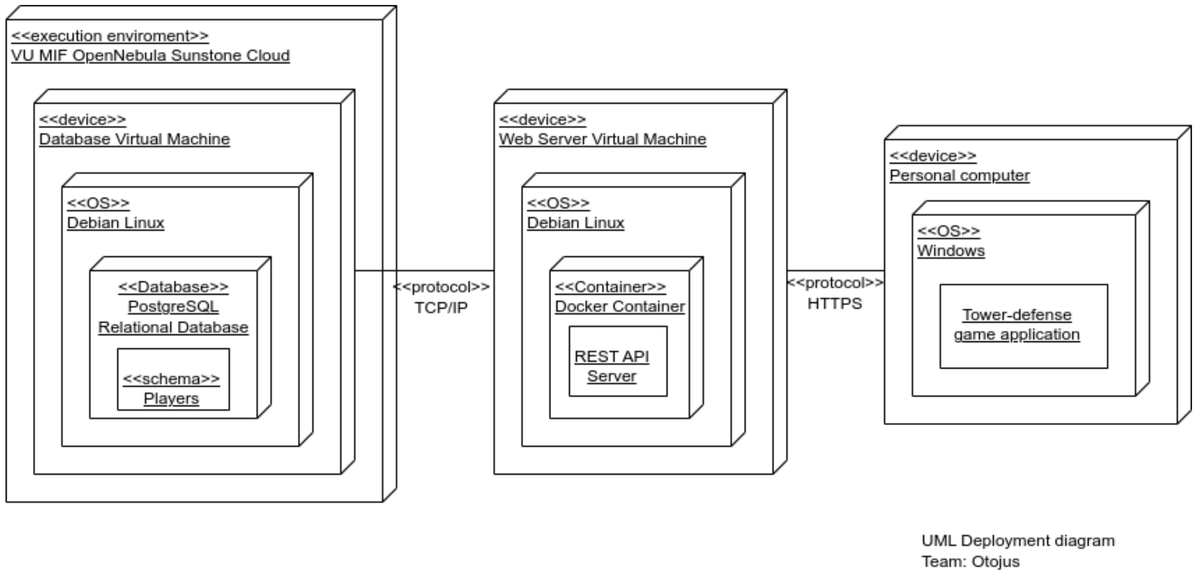
We will upload all the code made by us to our Git repository in GitLab

2 Diagrams that explain the system internals:

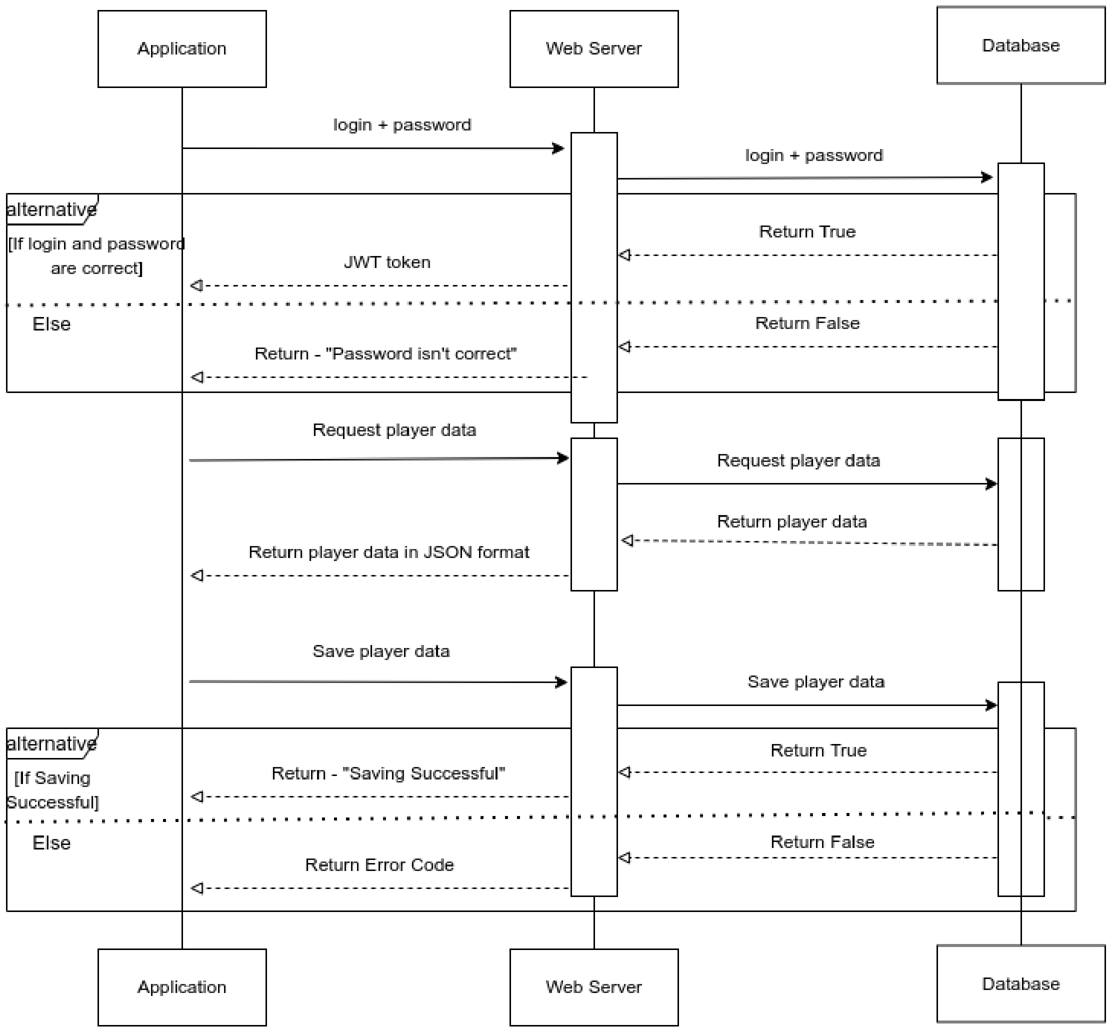
2.1 Action diagram - to show actions that can a player can perform in the tower defence game.)



2.2 UML deployment diagram - to show what artifacts your team's deliverable will consist of, and where they will be deployed/published



2.3 UML sequence diagram - to explain what are the main parts/modules /components and how they all connect together.



3 An overview of how we will approach testing:

For testing we will mostly do it manually with no automation. But for more later testing, a version of the game will be sent out for people to test. We are not going to automize back-end testing because we will have a simple database and webserver so we don't need to test it.

4 Detailed explanation on how everything works:

In the unity game - scripts: "GameManager" scripts controls the gameplay of the game (wave spawning and controlling which wave is being spawned with what enemies, enemies are spawned from a prefab(a prefabricated building), global variables that are used for health points, wave number, currency etc.)

Towers, Enemies and Projectiles have their own scripts that control their objective. Tower script gives the tower variables for damage and also detecting a circle size for the range of the tower, Enemy script directs the enemies to move to a given location and have health points that later own get damaged when colliding with a projectile that is spawned on a tower as a prefab(a prefabricated building) whenever the enemy goes into the tower range, projectile script is for targeting an enemies location and moving and checking if it collides with an enemy.

Webserver - One virtual machine has PostgreSQL database, webserver machine has connection to this machine by TCP/IP protocol. Clients have connection to the webserver by https protocol and send get requests to the webserver and webserver will get this information from the database and send it to clients.