In this thesis, we will develop an online cash register application for restaurants and events using a case study with Faros, a company part of the Cronos group. This application is a digital version of cash registers found in stores and restaurants. Customers will be able to use their own mobile devices to get an overview of the products available at a location and place orders. Employees will also use computers or their own devices to receive and process these orders. The application should be developed with distribution in mind and should function as a web application running in the cloud.

The core aspect of this project is to explore Java web technologies and analyse what features they offer. We research frameworks and libraries suitable for prototyping the case study, and defined constraints and other requirements with our coaches at Faros. We will focus specifically on distributed applications and the micro-services architecture. Multinational companies such as Netflix, Amazon and eBay are already running on micro-services and it is becoming more popular in the web development world. We will explore the advantages and disadvantages of this kind of architecture by developing the online cash register application. First as a single monolithic application and then rebuilt into a micro-service system which can run as several separate modules on multiple servers.

At the end of this project we should have an excellent understanding of the technologies involved to make this online cash register application that handles orders and payment in restaurants. It will also teach us the difficulties and advantages of implementing a micro-services system.

\\\\

We have explored different features of Spring Boot framework which is the second most popular web framework, and chose it to build our prototype. We switched from Monolithic architecture to Micro-services architecture which is a method of developing software applications as a suite of independently deployable, small, modular services. For each service, different database is used, with both MySQL and MongoDB. For each independent service application, an embedded Tomcat server from Intellij IDEA is used to deploy our project. Each service provides REST APIs for API Gateway and client-side web application, and both of which are configured as clients of Eureka discovery server that can detect devices and services registered on the Eureka registration server. In this case, scalability and load balance are guaranteed in this Micro-services architecture. An android application is made to consume the API Gateway.