Project Plan

Group 4: Payment gateway

Client: Okapi Finance

16.11.2017.

Table of contents

1. Introduction

1.1. Okapi Finance

Okapi represents fintech platform based on VISA, that provides virtual wallets and payment services to private and company customers. The main mission of Okapi finance is to bank unbanked by actively participating in financial inclusion on giving security, transparency, and compliance. Okapi meets all of its customers with heavy contractual feel. For the customers care and comfort Okapi has a built-in feature available in the system where Okapi customers can buy things from the companies that are registered with Okapi whereas, unregistered customers can also buy things from those companies that are aforementioned via using their VISA Card.

In the detailed view, Okapi allows both the registered and unregistered customers to use an Okapi as a gateway for their payment and sales. Moreover, to make an Okapi as your first choice Okapi provides a type of services where users can transfer their money without paying any additional charges than the actual amount. After the transaction approves a customer receives a final confirmation from the system for any type of clam that the users can make in future to the company. The main aim of Okapi is to Bank those who are unbanked by other means Okapi protect its customers from carrying a big amount of money all the time in their wallets which is not just difficult and heavy but could also cause a problem in the form of loss. Okapi focuses more on introducing people to up-to-date technologies and modern ways of life. In addition, a secure reliable, efficient, and guaranteed services of Okapi around the world has increasingly changed the mind of people to use an Okapi as their dreaming platform. This discussion proves that Okapi plays a significant role in the advancement of this era in order to make the people more familiar with the latest approach of technologies.

1.2. Payment Gateway

In general, Okapi releases the companies from spending more time and money on advertising to convince the people to use and buy their products. Okapi organization (webmasters/developers) has come up with a new idea to bring all of the web shops on a single platform for which they first need to be registered with okapi and should sign a contract for the specified limit of duration. Okapi mission is to provide a very simple and unique way to the customers to access more than many companies for their desired products. Okapi web server has the capacity to register and manage companies and customers on their joining request. However, they also have given the strength and potential to the companies to access their database and view the reports related to each customer on the same terms and conditions by them.

Okapi has introduced two ways of payment methods to the customers, Okapi’s wallet and the other one is any valid VISA account details that let the users make their desired wishes true and complete which clearly states that Okapi deals with its customers according to their demands and expectations. On the other side, Okapi also cut few transaction charges from the customer’s balance to make the performance of the service more perfect as well as these additional charges that are deducted from the user's account is to contribute to the other charity organizations. Okapi strategy behind the services is not just a business but also to make the people smile and happy with their lives.

2. Project organization

2.1. Project group

Nermin Imamovic (project manager), born on 08-01-1996 in Jablanica (Bosnia and Herzegovina). A master's student in Software Engineering. He finished primary and high school education in Sarajevo. After finishing high school education, he signed Faculty of Electrical Engineering at the University of Sarajevo where he finished his bachelor studies at Department of Computer Science and Informatics. Email: nimamovic9@gmail.com

Janne Suuronen (contact person), born on 20-03-1996. A third- year bachelor student in Computer Science. He finished primary school at Risbroskolan in Fagersta (Sweden) and went to high school at NVU Brinellskolan also in Fagersta.

Email: jsn15011@student.mdh.se

Arshad Ahmad, born on 20-10-1986. A master's student in Software Engineering. He got bachelor's degree in Telecommunication & Networking in 2015.

Email: myselfarshad.ak@gmail.com

Vadim Panin, born on 14-05-1990. A third-year bachelor student in Computer Science. He finished primary school at S:t Ilians and high school at Rudbeckianska in Västerås.

Email: fpn14001@student.mdh.se

Matthew Mccully, born on 07-09-1983. A third-year bachelor student in Computer Science. High school through Komvux Eskilstuna with a focus on programming and web development. Primary school through various institutes in California.

Email: matthew.mccully01@gmail.com

Darko Joška, born on 21-06-1993 in Cetinje (Montenegro). A master's student in Software Engineering. He finished primary and high school education in the town of Herceg Novi (his hometown). After finishing high school education, he moved to Podgorica (the capital city of Montenegro) where he finished his bachelor and specialization studies in Software Engineering. Email: darkojoska@gmail.com

Kostadin Rajković, born on 18-10-1994 in Niš (Serbia). A master's student in Software Engineering at Faculty of Sciences and Mathematics in Niš. He got bachelor's degree in Computer Science at the same faculty. He finished his primary and high school education also in Niš.

Email: kosta.rajkovic@gmail.com

2.2. Organization and communication

Being students we are unable to set up a fixed schedule from week to week. Due to the participants in this project studying different courses, communication will take place through various services such as Slack (a communication tool for professional developers). The project will be organized and updated through a shared repository on GitHub. Meetings between the group and between the customer will be handled on a weekly basis. Reporting work hours will be recorded in an external file on Google drive. Every day after meetings or working on a project, everyone will add working hours for himself. For organization and roles on the project, we will use a Trello board.

2.3. Planned effort

Because of the different schedule, everyone will plan their own time for working on this project. We have meetings with the steering group every Monday and with our client on Tuesdays. Besides that, we will have group meetings on a weekly basis to see if everything goes as planned. Our plan is to work on the project daily, rather than doing everything in one day before the deadline. Roughly, two hours will be the daily work on average. Of course, if there are things that require some additional time or some of us face problems, we will try to help each other and we will extend planned two-hour work if needed.

2.4. External and Internal deliverables

The external deliveries will consist of documentation, what we as a group have done and what we have achieved. This will then be sent to the steering group at every deadline in the form of a summary documentation of the project work.

Internal deliveries will consist of what all parties in the group have done and how it has been done. Each individual in the group will provide documentation of the things they have worked at. Internal deliveries will be delivered every Monday before the meeting with the steering group. Before the meeting with the steering group, we will have an internal meeting to review all the parts we have set deadlines for on Trello. Also to check if there are tasks that have not been completed yet. The internal delivery is on Mondays, also in order to be able to plan before the meeting with the client on Tuesdays. This allows us to discuss and show what progress we have made in the past week. In the case of late delivery, namely, if things have not been completed within the timeframe, we will review possible solutions to complete the delivery of a particular process. All deliveries within the group are presented in the form of documentation and; implementation of code and design.

2.5. Deadline and Milestone

Deadlines for the tasks will be every Monday, because of the meetings with the steering group.

- **The external deadline for the project planning is Thursday, November 16th.** The internal deadline on Thursday, November 16th.

- **The external deadline for the Design description and Product is Thursday, November 30th.**
  The internal deadline for this part of a project on Wednesday, November 29th.

- **The external and final deadline for the Project report and the Product is Thursday, January 11th.**
  The final internal project report and final product deadline are on Friday, December 22nd.

The milestone will be along with deadline where we will review all the work and see if we have managed to complete all the tasks. Completed tasks must be tested in order to see if they meet client requirements. A real test is carried out by the completed components of the project and brief documentation of each individual in the group about what has been done, how it works, possible advantages and disadvantages. This is then uploaded to a common group of files on the Google Drive.

2.6. Activities (meetings and presentations, presentations preparation, documentation)

Meetings will be at the beginning of a new iteration, which is on Mondays. On these meetings, we will discuss within the group what everyone has done. Presentation of the work will be done during the Monday meeting with the steering group. After the meeting, we will prepare the next iteration and review things that need to be done as well as presented before the meeting with the client on Tuesday.

Below, the Gantt chart is presented, as it is more easy to keep up with the project in that way.

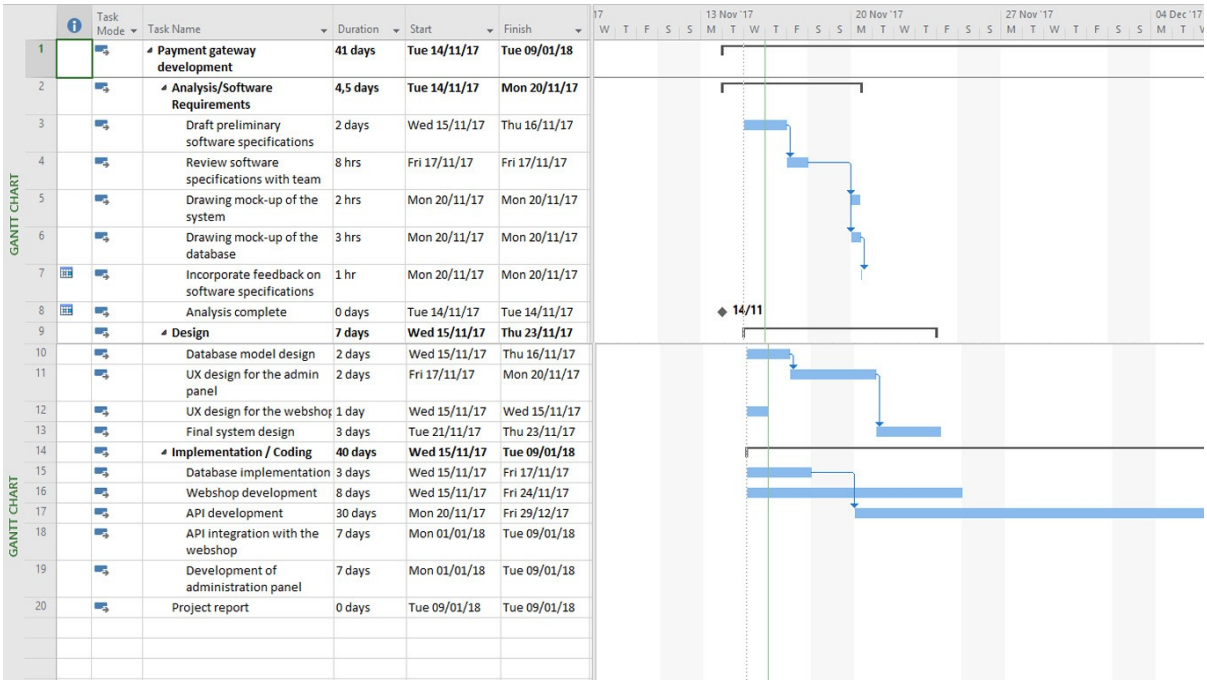

Figure 2.1: Gantt chart of the project

3. System description

3.1. Domain and problem

The domain related to the project is divided into two parts. First part is the web shop that the API is going to be integrated with. The webshop domain involves customers and the company deploying the webshop site and their respective intentions. However, these intentions are hard to generalize since they are very subjective depending on company and customer.

The second part of the overall domain for the project is the fintech domain since Okapi is a fintech company. The fintech domain involves a company deploying technological solutions to serve financial demands and to provide financial services. In this case, Okapi's main goal is to bank the unbanked. In the context of this project, Okapi wants to provide a financial solution for online payment. This solution is going to be used by companies in order to provide their customer's methods of paying on their web shops by using Okapi payment system.

With the domain explained, the problem the client wants to be solved is the following. Okapi wants to provide their customers a method of allowing them to use Okapi's wallet solution for e-commerce.

3.2. Existing systems

Currently, Okapi does not have an existing payment gateway which can be integrated into the web shops for its customers. The goal of this project is to implement a dummy system that can be integrated into their current database and provide an API which it's company customers can use to easily and safely add an Okapi payment gateway to their existing web shops.

Since Okapi does not have a currently implemented payment gateway it is our task to create one, and demonstrate to Okapi how it can be done and give them a base to build upon if they choose to implement our solution.

As said previously, they do not have a payment gateway, but that does not mean that we cannot incorporate work that they have done into our own project. One thing that could be of great value is their database. Their database contains information about customers that could be crossed over with the database we are creating. This would allow the use of currently entered customer information in the implementation of the payment gateway.

However, their customer database contains sensitive information, so in our implementation, one thing we can do is to create a similar dummy database and functions which target this structure. If Okapi chooses to incorporate our project into its existing system they can easily

make small changes to the code in order to target their currently existing database and its structure.

3.3. Desired functionality

To capture the desired functionality in the product. We constructed use case diagrams which visualize the intended ways of use for involved actors.

3.3.1. Roles in the system

Our system is divided into three parts, Okapi Payment Gateway API, a webshop for the demonstration and Administration panel. Our system consists of three roles, Global Administrator, Company Administrator, and Customer. Roles with their respective privileges for each part of the system are shown below.

Okapi Payment Gateway

Roles in the system	
Role	Privileges
Administrator	● Login to system ● Register companies ● Manage companies ● Register customers ● Manage customers ● View reports
Company	● Login to system ● View reports ● Configure API
Customer	● Pay with VISA ● Pay with OKAPI

Demonstration webshop

Roles in system	
Role	Privileges
Customer	● Browse product ● Add product to cart ● Checkout

Administrative webpage

Roles in system	
Role	Privileges
Administrator	• Login to system • Register companies • Manage companies • Register customers • Manage customers • View reports
Company	• Login to system • View reports • Configure API

3.4. Use Case Diagrams

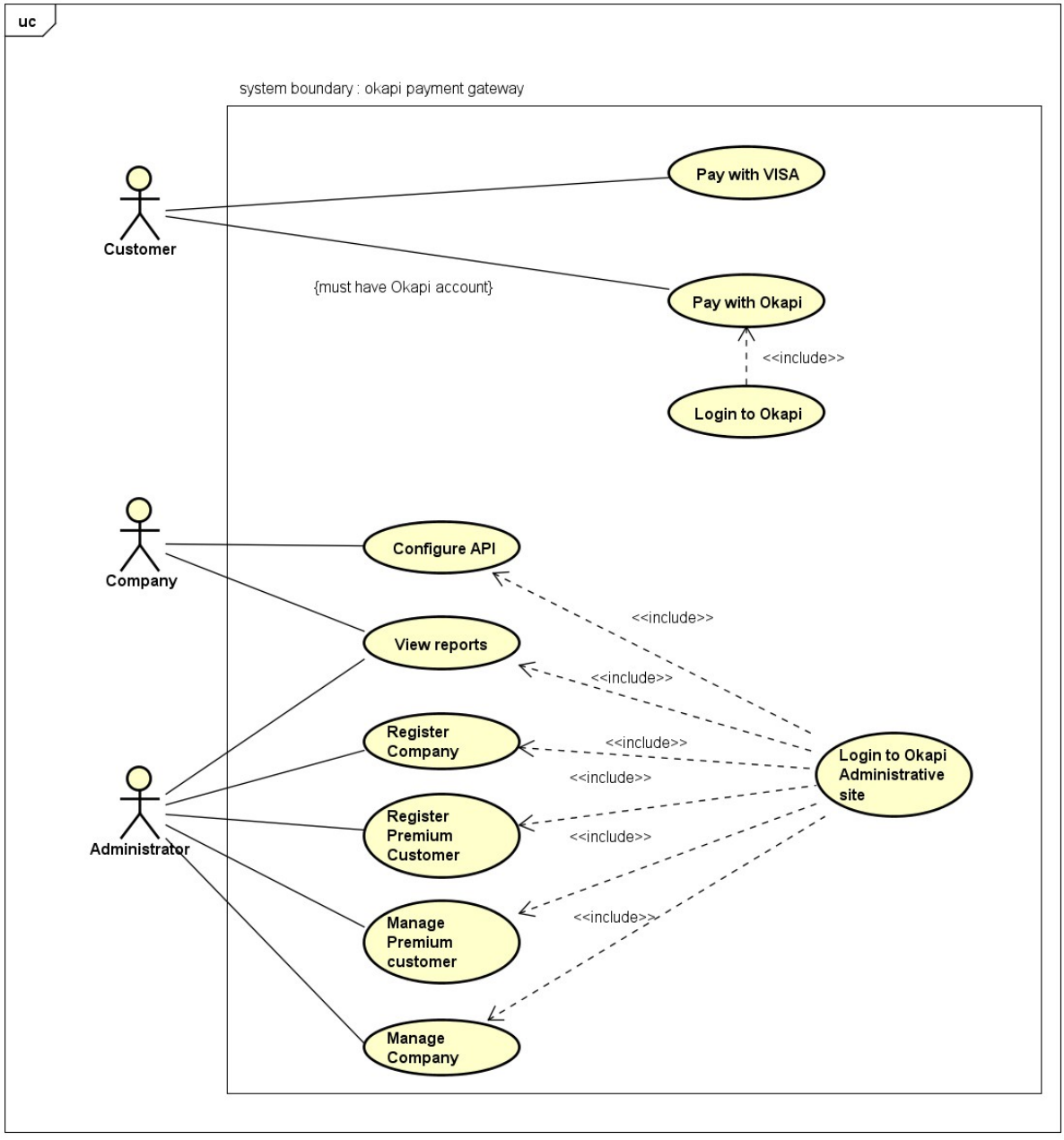

Figure 3.1: Use Case diagram for payment gateway

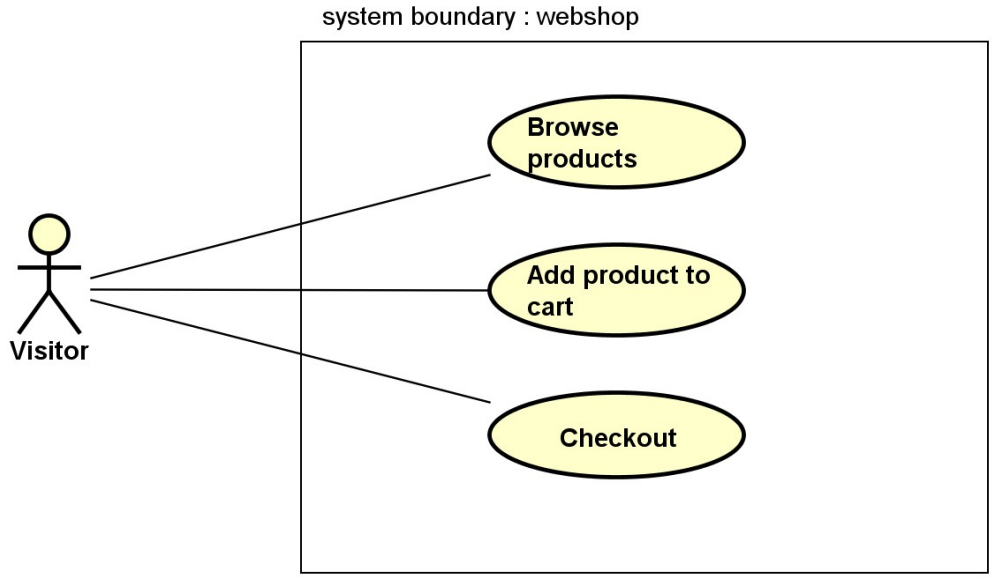

Figure 3.2: Use diagram for the demonstration webshop

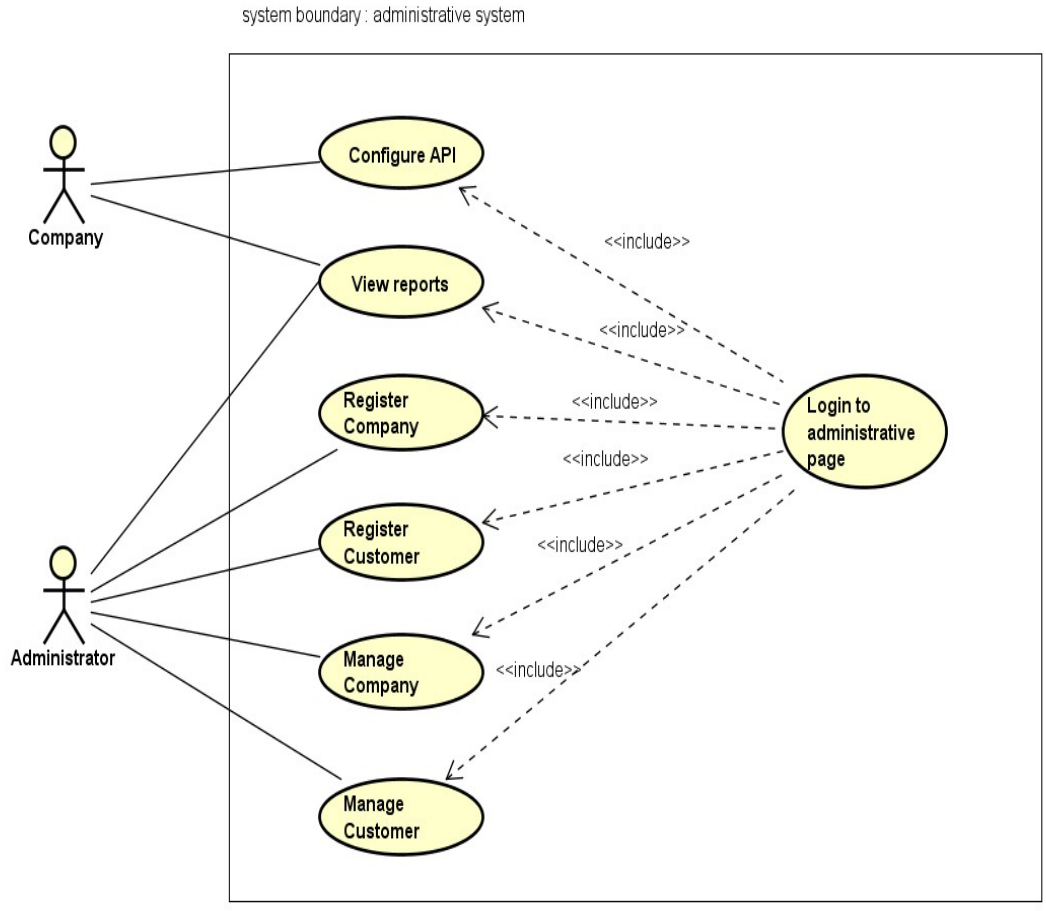

Figure 3.3: Use diagram for the Administrative web page

3.5. Functional Requirements

Okapi Payment Gateway

Login into administrative system	
Description	Administrator/Company logs into the Okapi system.
Actor/Actors	Administrator/Company
Preconditions	Administrator/Company must have Okapi Account
Main Scenario	1. Administrator/Company enters their user credentials. 2. System verifies credentials. 3. System logs in the administrator to the administration page
Extensions	1. Validation fails a)System displays that the validation failed and reason b)System allows retry 2. If user is company a)System logs company into the company-specific page

Register Company	
Description	An administrator can register a new company in the Okapi system.
Actor/Actors	Administrator
Preconditions	None
Main Scenario	1. Navigate to register new company page. 2. Enter the new companies details. 3. Validate inputted information. 4. Check company details against sanction lists. 5. The system registers the new company.
Extensions	1. Validation of details failed. a)Display that validation failed to the administrator and reason. b)The system allows for retry. 2. Company matched towards sanctions list a)Display that the sanction check failed. b)The system allows for retry.

Manage Company	
Description	The administrator manages the details of a registered company.
Actor/Actors	Administrator
Preconditions	Company to be managed must exist in system
Main Scenario	1. Navigate to manage company page. 2. Enter id for the company to be managed. 3. The system displays related details. 4. Administrator alters wanted details. 5. The administrator clicks “confirm changes”. 6. The system validates the changes. 7. System updates details.
Extensions	1. Identifier didn’t match a)Display to the administrator that the company could not be found. b)The system allows the user to input new company id. 2. Validation failed a)Display to the administrator that validation failed and reason. b)The system allows the administrator to try again.

Register Customer	
Description	An administrator can register a new customer in the Okapi system.
Actor/Actors	Administrator
Preconditions	None
Main Scenario	1. Navigate to register new customer page. 2. Enter the new customer details. 3. Validate inputted information. 4. Check customer details against sanction lists. 5. The system registers the new customer.
Extensions	1. Validation of details failed. a)Display that validation failed to the administrator and reason. b)The system allows for retry. 2. Company matched towards sanctions list a)Display that the sanction check failed. b)The system allows for retry.

Manage Customer	
Description	An administrator can register a new customer in the Okapi system.
Actor/Actors	Administrator
Preconditions	Customer must exist in system
Main Scenario	1. Navigate to manage customer page. 2. Enter id for the customer to be managed. 3. The system displays related details. 4. Administrator alters wanted details. 5. The administrator clicks “confirm changes”. 6. The system validates the changes. 7. System updates details.
Extensions	1. Identifier didn’t match a)Display to the administrator that the customer could not be found. b)The system allows the user to input new customer id. 2. Validation failed a)Display to the administrator that validation failed and reason. b)The system allows the administrator to try again.

View Reports	
Description	Administrator and/or Company can filter and view reports.
Actor/Actors	Administrator
Preconditions	Administrator/company user must be logged in.
Main Scenario	1. Navigate to the view reports page. 2. Enter company ID to get a report for. 3. The system generates a report for related company ID. 4. System displays generated a report.
Extensions	1. If user is company a) The system generates a report for the company. b) System displays generated a report. 2. Company ID didn’t match a) The system displays that ID could not be found. b) The system allows for retry.

Configure API	
Description	The company can configure the API for their webshop.
Actor/Actors	Company
Preconditions	Company user must be logged in
Main Scenario	1. Navigate to the configure API page. 2. Edit parameters currently in use.
Extensions	None

Pay with VISA	
Description	Customer pays with VISA Card
Actor/Actors	Customer
Preconditions	Customers must be logged in and have valid VISA card
Main Scenario	1. Navigate to the Payment page 2. User chooses VISA card as payment method
Extensions	1. Validation of details failed. a)Display that validation failed to the customer and reason. b)The system allows for retry. 2. Customer’s VISA card was not verified

Pay with OKAPI	
Description	Customer pays with OKAPI
Actor/Actors	Customer
Preconditions	Customers must be logged in and valid OKAPI account
Main Scenario	1. Navigate to the Payment page 2. User chooses OKAPI as payment method
Extensions	1. Validation of details failed. a)Display that validation failed to the customer and reason. b)The system allows for retry. 2. Customer doesn’t have OKAPI account a)System redirects customer for registration on OKAPI

Login to OKAPI	
Description	Customers log in to OKAPI account
Actor/Actors	Customer
Preconditions	Customers must be logged in and valid OKAPI account
Main Scenario	1. Navigate to OKAPI login page at website 2. User insert his username and password
Extensions	1. Validation of details failed. a)Display that validation failed to the customer and reason. b)The system allows for retry. 2. Customer doesn't have OKAPI account a)System redirects customer for registration on OKAPI

Demonstrative Webshop

Browse products	
Description	Using a search bar and/or other tools, visitor can browse products in the webshop
Actor/Actors	Customer
Preconditions	None
Main Scenario	1. Customer inputs keyword into the search bar 2. The system will display hits.
Extensions	None

Browse products	
Description	Add product to cart
Actor/Actors	Customer
Preconditions	None
Main Scenario	1. Visitor selects a product. 2. Clicks "add to cart" 3. System displays, "Product has been added to cart"

Extensions	1. Product is not in stock a)Add to cart button will be greyed out.

Checkout	
Description	The customer proceeds to the checkout and will see full details of the cart
Actor/Actors	Customer
Preconditions	None
Main Scenario	1. Visitor clicks proceed to checkout button 2. System will display everything in the cart 3. Visitor clicks continue to payment button
Extensions	1. Visitor edits the contents of the cart a)Cart is updated accordingly

Administrative website

Login into administrative system	
Description	Administrator/Company logs into the Okapi system.
Actor/Actors	Administrator/Company
Preconditions	Administrator/Company must have Okapi Account
Main Scenario	1. Administrator/Company enters their user credentials. 2. System verifies credentials. 3. System logs in the administrator to the administration page
Extensions	1. Validation fails a)System displays that the validation failed and reason b)System allows retry 2. If user is company a)System logs company into the company-specific page

Register Company	
Description	An administrator can register a new company in the Okapi system.
Actor/Actors	Administrator
Preconditions	None
Main Scenario	1. Navigate to register new company page. 2. Enter the new companies details. 3. Validate inputted information. 4. Check company details against sanction lists. 5. The system registers the new company.
Extensions	1. Validation of details failed. a)Display that validation failed to the administrator and reason. b)The system allows for retry. 2. Company matched towards sanctions list a)Display that the sanction check failed. b)The system allows for retry.

Manage Company	
Description	The administrator manages the details of a registered company.
Actor/Actors	Administrator
Preconditions	Company to be managed must exist in system
Main Scenario	1. Navigate to manage company page. 2. Enter id for the company to be managed. 3. The system displays related details. 4. Administrator alters wanted details. 5. The administrator clicks “confirm changes”. 6. The system validates the changes. 7. System updates details.
Extensions	1. Identifier didn’t match a)Display to the administrator that the company could not be found. b)The system allows the user to input new company id. 2. Validation failed a)Display to the administrator that validation failed and reason. b)The system allows the administrator to try again.

Register Customer	
Description	An administrator can register a new customer in the Okapi system.
Actor/Actors	Administrator
Preconditions	None
Main Scenario	1. Navigate to register new customer page. 2. Enter the new customer details. 3. Validate inputted information. 4. Check customer details against sanction lists. 5. The system registers the new customer.
Extensions	1. Validation of details failed. a)Display that validation failed to the administrator and reason. b)The system allows for retry. 2. Company matched towards sanctions list a)Display that the sanction check failed. b)The system allows for retry.

Manage Customer	
Description	An administrator can register a new customer in the Okapi system.
Actor/Actors	Administrator
Preconditions	Customer must exist in system
Main Scenario	1. Navigate to manage customer page. 2. Enter id for the customer to be managed. 3. The system displays related details. 4. Administrator alters wanted details. 5. The administrator clicks “confirm changes”. 6. The system validates the changes. 7. System updates details.
Extensions	1. Identifier didn’t match a)Display to the administrator that the customer could not be found. b)The system allows the user to input new customer id. 2. Validation failed a)Display to the administrator that validation failed and reason. b)The system allows the administrator to try again.

View Reports	
Description	Administrator and/or Company can filter and view reports.
Actor/Actors	Administrator
Preconditions	Administrator/company user must be logged in.
Main Scenario	1. Navigate to the view reports page. 2. Enter company ID to get a report for. 3. The system generates a report for related company ID. 4. System displays generated a report.
Extensions	1. If user is company a) The system generates a report for the company. b) System displays generated a report. 2. Company ID didn't match a) The system displays that ID could not be found. b) The system allows for retry.

Configure API	
Description	The company can configure the API for their webshop.
Actor/Actors	Company
Preconditions	Company user must be logged in
Main Scenario	1. Navigate to the configure API page. 2. Edit parameters currently in use.
Extensions	None

3.6. Non-functional requirements

Security

Security is the main aspect of this system and a lot of effort will be put into this part. There are some security regulations about VISA card like encryption of data and other secure transaction related concerns, such as not storing CVV numbers, hiding card numbers and other sensitive data. Users information also should be secured and encrypted.

Maintenance

The system should be well adjusted for maintenance, in the sense that it is needed to minimize time and influence on the overall work. To achieve this goal, all work on the system should be done on the time when the system is under significantly less load.

Usability

GUI design will follow modern and the most usable trends from the field of UX/UI design. This way users can effectively and efficiently do their tasks. It also enables easy usage for people who are not professionals when it comes to using computers.

Portability

End users access the system as a web application. This ensures high portability by providing customers to access it on a wide variety of devices. The most important thing on these devices are installed supported web browsers, and that an internet connection is available.

Scalability

The definition of scalability is the ability of the system to adapt to increased processing requirements in a predictable way without the system becoming too complex, expensive and impractical. With a good system, the client will be able to add new desirable functionalities and increase the number of users easily. Increasing the number of users and requests will not reduce the response time of the system, and the system will work efficiently.

4. Initial project backlog

This is a list of the system's features to be implemented (from highest to lowest priority):

- Okapi Payment Gateway
  - API

    Our highest priority is to develop API in a way that company's as Okapi's associates could easily integrate with their existing webshops and that will handle the payment process. This part of the system will be invisible to the end-users and it will be triggered only when they choose payment button from the webshop. It is supposed to create transaction contract and get transaction acknowledgment.

    Estimated working time: 30 days

  - Payment Page

    Second highest priority is to develop payment gateway for Okapi. It is supposed to be some kind of a popup window or a separate webpage that will handle payments for Okapi users. It has to be secure and safe from malicious users on the internet and it also has to take care of user's privacy. This is the most important part of the system for the end-users since they will see it every time they are making payment to a webshop by using Okapi platform. We should implement two payment options on the payment page, for Okapi and Visa users. Okapi users will enter their Okapi credentials, while Visa users will enter their card number. Payment page should show payment details and then execute the payment.

    Estimated working time: 30 days (along with API)

  - Admin Panel

    Admin should be able to log in to the system and then register and manage companies and customers and view reports and all relevant data as well. Admin panel has medium priority.

    Estimated working time: 7 days

  - Company Panel

    The company representative should be able to login to the system and view the reports related to his company and configure API for his webshop. Company panel has medium priority.

    Estimated working time: 7 days (Along with Admin panel)

  - Backend Logic/Business

    Backend logic is mainly related to communication with Okapi's database and secure way of updating data tables. The database will be used for storing user data and transactions. For this purpose, we will implement an imitation of Okapi's database that will have only necessary tables for a demonstration of the system. Since Okapi's

original database is developed in MsSQL it is required that our imitation of the database is also developed in MsSQL.

Estimated working time: 7 days

- Web Shop

Webshop will be used only for a simulation of triggering payment process and making the transaction between Okapi and a webshop company. Any modern technology for websites development is allowed. This is the part of the system with the lowest priority since in reality any webshop could be used and it is not the essence of the project. It should have a shopping cart and implement API of Okapi Payment Gateway.

Estimated working time: 8 days