

# Design document

Jack Mutsers  
Fontys ICT  
Software engineering semester 3  
Avetyan,Rafayel R.  
Franssen,Michael M.G.J.

## Index

Versions table.....	3
Front-End Framework Selection.....	4
Features.....	4
Conclusion .....	4
Back-End Framework Selection.....	5
Features.....	5
Conclusion .....	5
security related design decisions .....	6
Authentication.....	6
Authorization.....	6
Password security.....	6
Potential OWASP security flaws.....	7
CI Setup.....	8
Description: .....	8
Visual representation: .....	8

## Versions table

Version	changes
1.0	Created the document with the front-end framework selection
2.0	Added the back-end framework selection
3.0	Added security related design decisions + title page, index and versions table

## Front-End Framework Selection

- main research question** what kind of libraries and frameworks could be used to help self-employed people to create invoices in such a way that it's easy to use, reusable and maintainable?
1. what methods could be used to create a simple yet effective way to create an invoice?
  2. what methods could be used to create a reusable invoice template that also has the potential to be updated?
  3. what framework could be used to implement the required methods and features
  4. what libraries for the chosen framework could help with the implementations of the required features

### Features

- drag and drop** this way the user can create a template by adding a elements to the page and drag it the location that they want the information to be on
- rest service client** this is needed to be able to manipulate the data on the database, that both the companies and the customers can connect to.
- Multipage** this is needed because several crud functionalities are present in the application, and if it's all on one page than it becomes messy and very un-user friendly
- form validation** this is needed to avoid sending incomplete or incorrect information to the api, which might cause problems
- pdf conversion** this is needed, because the companies and the customers must be able to download + print the invoices that are created

	React	Angular	Vue
drag and drop	+	+	+
rest service client	+	+	+
multipage	-	+	+
form validation	+	+	+
pdf conversion	+	+	+
	4	5	5

### Conclusion

both Vue js and angular are capable of all the required functionalities, but from what i could see angular seems to be a bit more easy to use for my required features . The coding structure of both frameworks are close, but based on the examples that i could see of the required features it seems that angular is just bit more manageable and clean. that is why i have chosen angular as the framework for my personal project.

## Back-End Framework Selection

<b>main research question</b>	what kind of libraries and frameworks could be used to help self-employed people to create invoices in such a way that it's easy to use, reusable and maintainabl
<b>1.</b>	what methods could be used to create a simple yet effective way to create an invoice?
<b>2.</b>	what methods could be used to create a reusable invoice template that also has the potential to be updated?
<b>3.</b>	what framework could be used to implement the required methods and features
<b>4.</b>	what libraries for the chosen framework could help with the implementations of the required features

## Features

<b>drag and drop</b>	this way the user can create a template by adding a elements to the page and drag it the location that they want the information to be on
<b>rest service client</b>	this is needed to be able to manipulate the data on the database, that both the companies and the customers can connect to.
<b>Multipage</b>	this is needed because several crud functionalities are present in the application, and if it's all on one page than it becomes messy and very un-user friendly
<b>form validation</b>	this is needed to avoid sending incomplete or incorrect information to the api, which might cause problems
<b>pdf conversion</b>	this is needed, because the companies and the customers must be able to download + print the invoices that are created

	C#	Java
database framework	+	+
restfull api	+	+
password encryption	+	+
authenticatie services	+	+
file upload support	+	+
	5	5

## Conclusion

Both C# and Java offer all the components that are required to build my application, that is why after some thinking I've decided to choose Java for my project. the reason for the decision is that I would like to learn how to program into Java, so that I have some experience with both C# and Java for the next time i need to make a decision like this. another reason is that Java is something we have to use for the current semester, because of this it will be easier to ask for help in case I get stuck and in turn will allow me to learn and get a better understanding about the programming language faster.

## security related design decisions

### Authentication

I have decided to make use of a login system to generate a JWT Token that has to be used for all Requests that are being made of the Restfull API service

### Authorization

For the Authorization I make use of user roles to dictate who can access what restful service he/she can use. Whenever a user makes a request the service with validate the JWT Token + user permissions to decide whether or not the user is granted access.

### Password security

To make sure that the passwords are safely stored in the database, I make use of BCrypt to encode and validate the user passwords, so that the password is never visible, not even in the database. Furthermore when the user makes a login attempt than the username + password are encoded in a base64 encoding to make sure no one can just see what credentials are being used.

## Potential OWASP security flaws

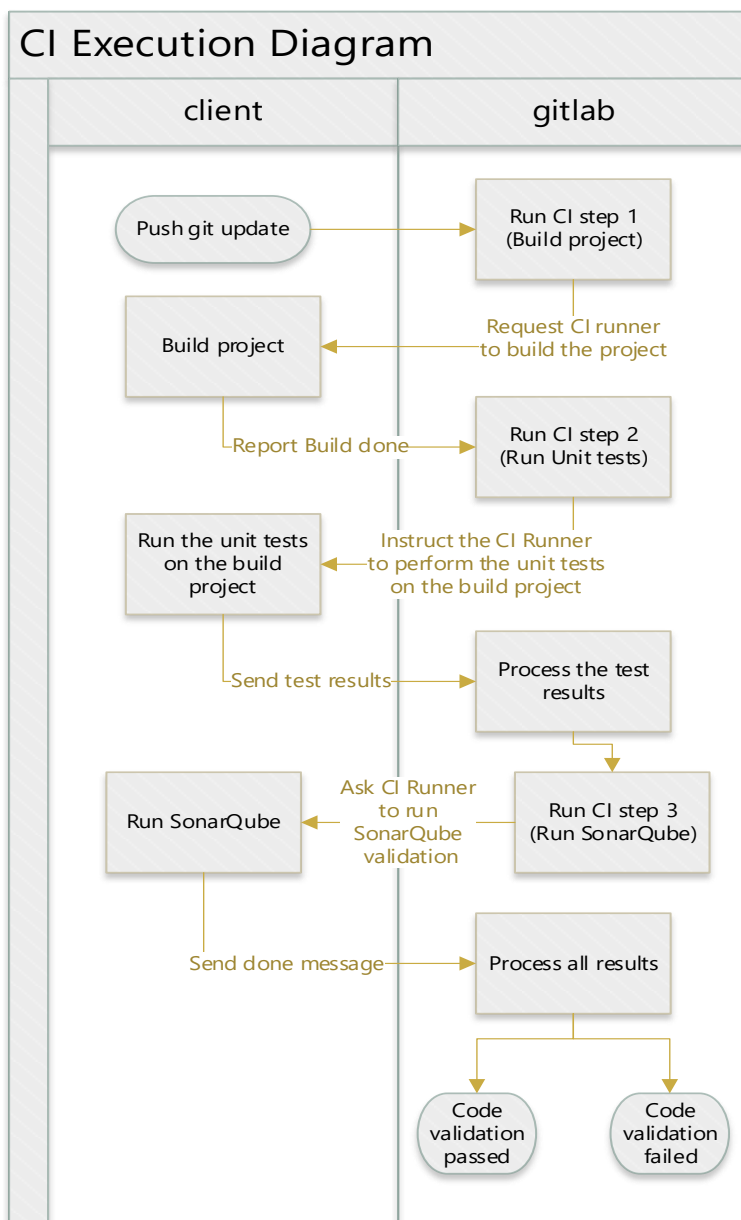
Issue	description	Potential solution
Broken Authentication	The JWT token could be intercepted + the login data is easy to decipher if someone intercepts the login request of a user	Implementing a SSL certificate + implementing a more secure validation that base64 to encrypt the login credentials
Broken Access Control.	The users might be able to alter some data for other users by modifying the json string that is being send to the restfull api	Perform a extra validation using the JWT Token to validate if user is who he says he is and if so implement extra user rights validations.)
Cross-Site Scripting XSS	A hacker might be able to commit JavaScript functions to the database for malicious intentions	Character validation on when updating/adding records for special characters and convert them to their UTF-8 version.
Insufficient Logging & Monitoring	If a system does not log anything than hackers can attack the system as much as they want and they wont be found out	Implement a logger on all restfull services + implement account disable on X amount of incorrect login attempts + ip address blocks on repeated authentication errors

## CI Setup

### Description:

At the start of the project I decided to upload the project to gitlab, and implement a CI integration. During the setup I configured my CI so that when I commit my code that the CI would build my project, run my implemented unit tests and validates my code quality based on the code diagnostics that I get from an external tool named SonarQube.

### Visual representation:





# Architecture Diagram

## System context

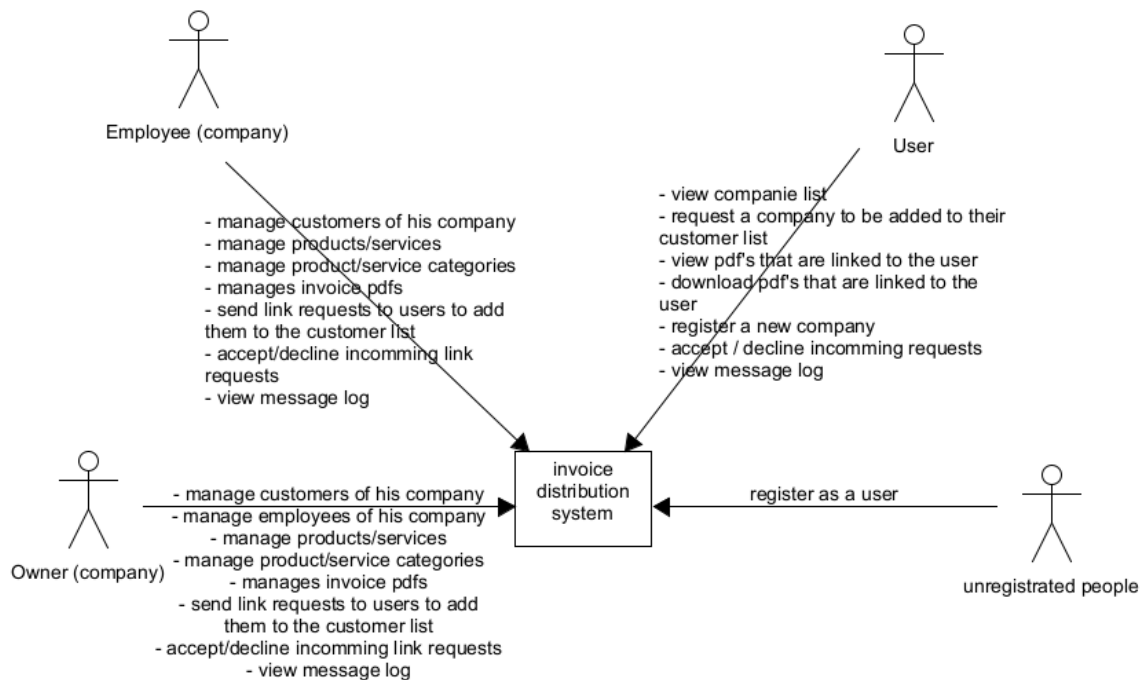
### Who uses the system

- Companies that want an easy way to share and create invoice pdf's
- Self employed workers that want an easy way to share and create invoice pdf's
- Customers that want an easy way to find and search previous invoices

### What other systems are linked to my application

besides my own web-application with its back-end restful api system do I not connect to any external systems.

## Use case diagram

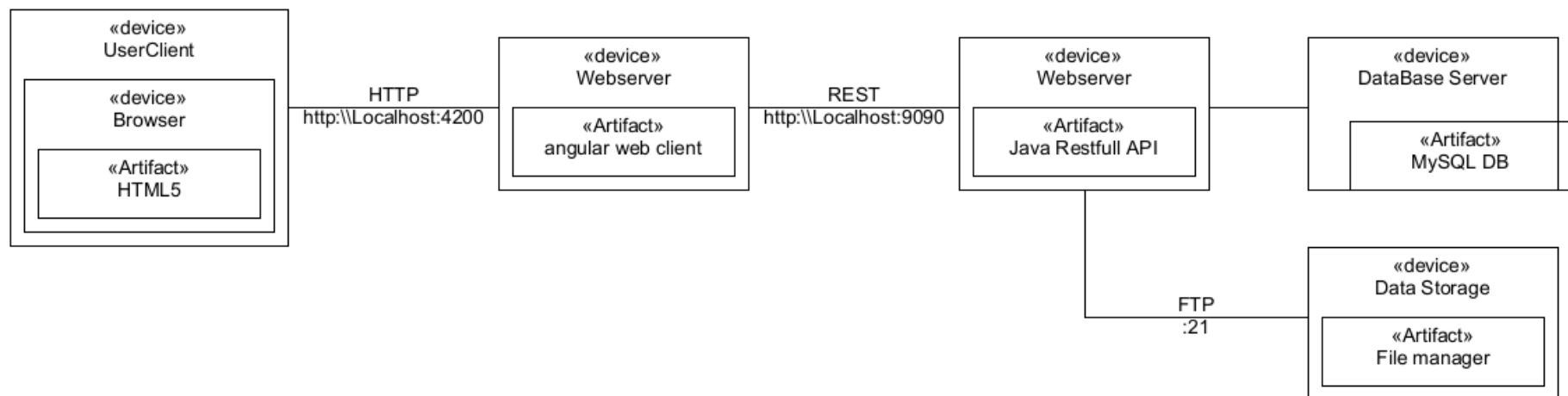


## Containers and technology

### Distributed system

- User Client
- Client server
- Restful API server
- Remote database
- Remote file storage

### Deployment Diagram



## Class diagrams

Please look at the added files in the class diagrams folder.