

# EWA TECHNICAL DOCUMENTATION

Simon Vriesema, Maajid Saidy, Dennis Moes, Ilias Achbal, Frank Oud  
HVA Amsterdam

## Table of contents

<b>1) Introduction .....</b>	<b>2</b>
<b>2) Product vision .....</b>	<b>3</b>
<b>3) Summary of most important epic stories .....</b>	<b>4</b>
1. As a specialist I want to have a profile .....	4
2. As a user I want to be able to login .....	5
3. As a client I want to be able to start a new project .....	6
4. As a Specialist I'd like to have an account .....	7
5. As a Specialist I'd like to manage my assigned projects .....	7
6. As a Client I'd like to have an account .....	7
7. As a Client I'd create projects .....	7
8. As a Client I'd manage projects .....	7
9. As an Admin I'd like to be able to manage all data of the application. ....	7
<b>4) Layered Architecture Package Diagram .....</b>	<b>8</b>
<b>5) Navigable Class Diagram .....</b>	<b>11</b>
1. Model directory .....	11
<b>6) The Challenges and Alternative Solutions .....</b>	<b>12</b>
<b>7) Deployment Diagram .....</b>	<b>13</b>
<b>8) Analyzing Design Performance .....</b>	<b>14</b>

## 1) Introduction

This document is intended for the deployment engineers or future software engineers or architects who get the task to extend your product with additional functionality or migrate it to another IT technology. The purpose of this document is to provide a high-level overview of the different technologies that are available for deploying and running your product, and to help you choose the right one for your specific needs. This document is not a comprehensive guide to using all the different technologies, but it does include detailed information on a few of the most popular ones. The target audience for this document is deployment engineers or future software engineers or architects who are looking to select the right technology for their product and are not familiar with all the available options.

## 2) Product vision

We are constantly looking for ways to improve our platform and create a product that is tailored to the needs of our clients. We are committed to staying up to date with the latest technologies and trends to ensure that our platform is always ahead of the curve. We strive to create a platform that is comprehensive and easy to use, providing our clients with the tools and resources they need for successful software development projects.

We understand that software development projects can be complex, and thus we are dedicated to making the process as smooth and stress-free as possible. Our goal is to create a platform that connects our clients with the best possible specialists to help them achieve their software development goals.

We are confident that our platform will provide Florijn's clients with the best possible experience and allow them to easily find the right specialists for their projects. We are committed to making sure our platform is always up to date and providing our clients with the most secure and efficient platform possible

### 3) Summary of most important epic stories

Epic stories are used in software development to help organize and structure projects. In this context, they are often used to describe the overall goals of a project and to serve as a template to break down the tasks into smaller stories.

#### 1. As a specialist I want to have a profile

As a specialist, I want to have a profile that I can create and manage on the application. This profile will allow me to store and display my skills, personal information such as my name, address, phone number, and email, as well as upload my Curriculum Vitae. This will allow me to showcase my qualifications to others and provide them with a way to easily contact me. The profile should also provide me with an easy way to edit and update this information as necessary. The profile will be used to authenticate myself when accessing the application.

Quality Attribute Descriptions:

- **Functionality:** The system shall provide all the necessary functionality for the specialist to manage, view and update their profile with their personal contacts.
- **Usability:** The system shall provide an easy-to-use interface that allows the specialist to perform all the necessary actions related to their profile efficiently and with minimal error rates.
- **Security:** The system shall implement appropriate security controls, such as authentication, to ensure that only authorized users can access the profile data.
- **Data Integrity:** The system shall maintain the authenticity and validity of the information within the specialist's profile, ensuring that it is consistent and accurate.
- **Maintainability:** The system shall provide the capability for the specialist to update their profile information as required, and for the system administrator to maintain the profile records with minimal effort.
- **Performance:** The system shall provide fast and responsive access to the specialist's profile, even when handling a high number of requests.
- **Accessibility:** The system shall be accessible for all users including those with disabilities, ensuring that the specialist profile can be created, edited, and accessed by anyone.

## 2. As a user I want to be able to login

As a user, I want to have the ability to log into the application, in order to access my profile and other features that are only available to logged-in users. The login process should be simple, secure, using standard authentication methods such as email and password. I should also have the ability to reset my password if I ever forget it. This login feature will provide me with a convenient and secure way to access my profile and other features.

### Quality Attribute Descriptions:

- **Functionality:** The system shall provide the functionality for the user to log in and access their profile and other exclusive features.
- **Security:** The system shall implement appropriate security controls such as standard authentication methods like email and password, to ensure that only authorized users can access the system.
- **Usability:** The system shall provide an easy to use and intuitive interface for the user to log in to the system, with clear guidance.
- **Maintainability:** The system shall provide the capability for the user to reset their password if ever forgotten.
- **Availability:** The system shall have a high availability ensuring that users can login to the system at any time.
- **Privacy:** The system shall protect the user's personal and login information and keep them confidential.

### 3. As a client I want to be able to start a new project

As a client, I want to have the ability to start a new project on the application, to initiate and manage projects efficiently. This feature should allow me to create a new project and provide basic information such as its name, description, and deadline. Additionally, I should be able to specify any special requirements or details about the project. Once the project is created, I should be able to assign specialists or invite them to apply for the project. This feature should provide me with a simple and efficient way to initiate a new project and manage its progression.

#### Quality Attribute Descriptions:

- **Functionality:** The system shall provide the capability for the client to create and manage projects, including inputting basic information such as name, description, and any special requirements.
- **Usability:** The system shall provide an easy-to-use and intuitive interface for the client to initiate and manage projects, with clear guidance and minimal error rates.
- **Maintainability:** The system shall provide the capability for the client to edit and update the project information and invite specialists or assign them to the project as necessary.
- **Data Integrity:** The system shall maintain the authenticity and validity of the project information, ensuring that it is consistent and accurate.
- **Performance:** The system shall provide fast and responsive access to the client's project list even when handling a high number of requests.
- **Security:** The system shall implement appropriate security controls to ensure that only authorized users can access the projects information and assign them to the project.

#### 4. As a Specialist I'd like to have an account

As a specialist I'd like to be able to create an account where I can save all my skills and personal data like name, address phone number, email, and Curriculum Vitae. which I'd like to be able to see and edit in the future. This account should also be used to authenticate when trying to access the application.

#### 5. As a Specialist I'd like to manage my assigned projects

As a specialist I'd like to be able to manage my assigned projects, this means being able to see which projects have been assigned to me and what announcements have been made within that project together with contact details of the project owner.

#### 6. As a Client I'd like to have an account

As a client I'd like to be able to create an account where I can save all personal data like name, address phone number, email. which I'd like to be able to see and edit in the future. This account should also be used to authenticate when trying to access the application.

#### 7. As a Client I'd create projects

As a client I'd like to create projects this means creating new projects and assigning which skills are required for the specialists and extra clarifying data.

#### 8. As a Client I'd manage projects

As a client I'd like to manage projects, this means being able to see a list of all my projects and filtering through them. And being able to see all the data and specialists for each project and potentially also make announcements.

#### 9. As an Admin I'd like to be able to manage all data of the application.

As an admin I'd like to manage all data of the application, this means being able to create, read, update, and delete all entities of the application meaning the specialists and their approval status, clients, projects, and even other admins.



## 4) Layered Architecture Package Diagram

Layered Architecture Package Diagram of our solution and elaborate on relevant choices and challenges within your logical architecture.

Presentation Layer:

- Package: frontend/src

- Classes:

- ClientDetail.vue, ClientList.vue, CreateClient.vue, AddSpecialistToProject.vue, CreateNewProject.vue, projectEvent.vue, projectSubmissions.vue, projectSubmissionsDetail.vue, CreateSpecialist.vue, Speecialists.vue, dashboard.vue, profile.vue, SpecialistHomePage.vue, SpecialistPendingPage.vue, SpecialistProfile.vue, SpecialistProjectsOverview, SpecialistSettings.vue, ClientHomePage.vue, ClientProfile.vue, ClientSettings.vue, createProjects.vue, landingPage.vue, Login.vue, ProjectsOverview.vue, RegisterPage.vue

- Interfaces:

- availableHours.vue, calender.vue, ProjectCard.vue, skillItem.vue, skills.vue, editProjectStatus.vue, projectCreateNew.vue, AdminDetail.vue, AdminsTable.vue, CreateAdmins.vue, BarChart.vue, LineChart.vue, PieChart.vue, FileList.vue, FileUpload.vue, AttachmentRow.vue, AvailabilityCard.vue, AvailabilityRow.vue, CalenderRow.vue, MeetingType.vue, SkillBadge.vue, UpcomingMeeting.vue, Sidebar.vue, SidebarLink.vue, LoadSpinner.vue, ProjectCard.vue, EditUserInfoModal.vue, SkillModal.vue, UpcomingMeetingModal.vue, CustomModal.vue, 404-page.vue, ClientNavbar.vue, Footer.vue, HelloWorld.vue, PageHeader.vue, SpecialistApplicationModal.vue, SpecialistNavbar.vue, SpecialistsTable.vue

- Dependencies:

- "@fortawesome/fontawesome-free": "^6.2.0",  
"@headlessui/vue": "^1.7.3",  
"@heroicons/vue": "^2.0.11",  
"@iconify/vue": "^4.0.0",  
"@popperjs/core": "^2.11.6",  
"animate.css": "^4.1.1",  
"axios": "^1.2.0",  
"chart.js": "^3.9.1",  
"chartist": "^1.3.0",  
"core-js": "^3.8.3",  
"crypto-js": "^4.1.1",  
"dateformat": "^5.0.3",  
"flowbite": "^1.5.3",  
"flowbite-vue": "^0.0.6",

## Technical documentation

```
"http-server": "^14.1.1",
"v-calendar": "^3.0.0-alpha.8",
"vue": "^3.2.13",
"vue-chartist": "^3.0.0",
"vue-chartjs": "^4.1.2",
"vue-datepicker-next": "^1.0.2",
"vue-final-modal": "^3.4.11",
"vue-router": "^4.1.5",
"vue-toastification": "^2.0.0-rc.5",
"yarn": "^1.22.19"
"@babel/cli": "^7.19.3",
"@babel/core": "^7.20.5",
"@babel/eslint-parser": "^7.12.16",
"@babel/preset-env": "^7.20.2",
"@vue/cli-plugin-babel": "~5.0.0",
"@vue/cli-plugin-eslint": "~5.0.0",
"@vue/cli-plugin-unit-jest": "^5.0.8",
"@vue/cli-service": "~5.0.0",
"@vue/test-utils": "^2.2.6",
"@vue/vue3-jest": "^27.0.0",
"autoprefixer": "^10.4.13",
"babel-jest": "^27.5.1",
"eslint": "^7.32.0",
"eslint-plugin-vue": "^8.0.3",
"jest": "^27.5.1",
"postcss": "^8.4.19",
"tailwindcss": "^3.2.3",
"vue-cli-plugin-tailwind": "~3.0.0",
"vue-cli-plugin-vuetify": "^2.5.8"
```

### Business Logic Layer:

- Package: com/hva/helios/rest, com/hva/helios/notifications
- Classes:
  - AnnouncementDistributor, NotificationDistributor, AnnouncementController,
  - AuthorizationController, EmailController, FileController, HourController,
  - JWTRequestFilter, ProjectController, SkillController, UserController
- Interfaces: none
- Dependencies:
  - org.springframework.boot:
    - spring-boot-starter-parent, spring-boot-starter-web, spring-boot-starter-test,
    - spring-boot-starter-data-jpa, spring-boot-starter-mail, spring-boot-starter-websocket

## Technical documentation

- io.jsonwebtoken:  
jjwt-api, jjwt-impl, jjwt-jackson
- com.h2database: h2
- org.jetbrains: annotations

### Data Access Layer:

- Package: com/hva/helios/repositories

- Classes:

AvailableHourJpaRepository, UserSkillJpaRepository,  
AnnouncementRepository, HourJpaRepository, ProjectJpaRepository,  
SkillJpaRepository

- Interfaces:

AdminJpaRepository, ClientJpaRepository, EventJpaRepository,  
SpecialistJpaRepository, UserJpaRepository, EntityRepository, testRepo,  
FileJpaRepository

- Dependencies: None

Navigable Class Diagram of the functional model (of domain entities) within your solution. Also create a class diagram that explains the structure and dependencies of one of the repositories in the JPA persistence layer. Also show a class diagram that explains the use of an external interface. Clarify all diagrams with relevant text.

```

classDiagram
    class EventType {
        +values() EventType[]
        +valueOf(String) EventType
    }
    class Event {
        +associateSpecialist(Specialist) boolean
        +associateProject(Project) boolean
        +dissociateSpecialist(Specialist) boolean
        +dissociateProject(Project) boolean
        +description String
        +project Project
        +start Date
        +location String
        +title String
        +id Long
        +eventType EventType
        +accepted int
        +user Specialist
        +end Date
    }
    class Hour {
        +hourEnd Date
        +available boolean
        +id Long
        +hourStart Date
        +label String
    }
    class AvailableHour {
        +days Map<String, Hour>
        +id Long
    }
    class Skill {
        +name String
        +projects Set<Project>
        +id Long
    }
    class UserSkill {
        +dissociateSpecialist(Specialist) boolean
        +associateSpecialist(Specialist) boolean
        +specialist Specialist
        +level Skill
        +skill Skill
        +id Long
    }
    class Specialist {
        +addSkill(UserSkill) void
        +removeSkill(long) void
        +dissociateUserSkill(UserSkill) boolean
        +associateUserSkill(UserSkill) boolean
        +associateEvent(Event) boolean
        +addProject(Project) boolean
        +removeProject(long) boolean
        +dissociateEvent(Event) boolean
        +approvalStatus long
        +projects Set<Project>
        +hours AvailableHour
        +available int
        +specialistType String
        +events Set<Event>
        +skills Set<UserSkill>
        +id Long
    }
    class Client {
        +website String
        +id long
    }
    class Admin {
        +id long
    }
    class User {
        +specialist Specialist
        +bio String
        +client Client
        +phone String
        +second_name String
        +address String
        +last_name String
        +password String
        +first_name String
        +email String
        +city String
        +zipCode String
        +photo String
        +admin Admin
        +userType Long
        +id long
    }
    class Project {
        +removeSkill(long) boolean
        +addSpecialist(Specialist) boolean
        +addSkill(Skill) boolean
        +removeSpecialist(Specialist) boolean
        +associateEvent(Event) boolean
        +dissociateEvent(Event) boolean
        +removeSkill(Skill) boolean
        +removeSpecialist(long) boolean
        +name String
        +description String
        +bannerUrl String
        +id Long
        +created Date
        +events Set<Event>
        +status int
        +skills List<Skill>
        +specialists Set<Specialist>
        +user User
    }
    class Announcement {
        +project Project
        +dateTime LocalDate
        +id Long
        +message String
        +user User
    }
    class FileModel {
        +data byte[]
        +name String
        +userId long
        +type String
        +timestamp LocalDateTime
        +id String
    }
    class JWTToken {
        +encode(String, String, int, long) String
        +decode(String, String, String) JWTToken
        +callName String
        +accountid Long
        +userType Long
    }
    class LoginBody {
        +email() String
        +password() String
    }
    class LoginResponse {
        +approvalStatus() Long
        +id() Long
        +userType() Long
    }

    EventType "1" -- "*" Event : event type
    Event "1" -- "*" Specialist : specialist
    Event "1" -- "*" Project : project
    Event "1" -- "*" Hour : hour
    Event "1" -- "*" Skill : skill
    Event "1" -- "*" UserSkill : user skill
    Event "1" -- "*" Announcement : announcement
    Hour "1" -- "*" AvailableHour : create
    AvailableHour "1" -- "*" Hour : update
    Skill "1" -- "*" UserSkill : update
    UserSkill "1" -- "*" Skill : update
    Specialist "1" -- "*" Hour : create
    Specialist "1" -- "*" AvailableHour : update
    Specialist "1" -- "*" UserSkill : update
    Specialist "1" -- "*" Project : update
    Specialist "1" -- "*" Announcement : update
    Client "1" -- "*" User : client
    Admin "1" -- "*" User : admin
    User "1" -- "*" Project : project
    User "1" -- "*" Announcement : announcement
    Project "1" -- "*" Specialist : specialist
    Project "1" -- "*" Skill : skill
    Project "1" -- "*" UserSkill : user skill
    Project "1" -- "*" Announcement : announcement
    Announcement "1" -- "*" Project : project
    Announcement "1" -- "*" User : user
    FileModel "1" -- "*" User : file model
    JWTToken "1" -- "*" User : jwt token
    LoginBody "1" -- "*" User : login body
    LoginResponse "1" -- "*" User : login response
  
```

## 6) The Challenges and Alternative Solutions

When designing our project for Florijn, we faced the challenge of designing a highly secure login system for our platform that handles sensitive personal information. Security was a key concern and we conducted extensive research to evaluate various options to ensure the safety of users' data.

After evaluating different alternatives, we decided to implement JSON Web Tokens (JWT) for authentication. JWT is a popular open standard that enables secure communication between parties by allowing them to share JSON objects in a compact, URL-safe format. This approach involved the server creating a JWT upon successful login and sending it to the client, which would then include it in subsequent requests to the server to prove its authenticity. This approach proved a more secure and scalable solution than using traditional username and password without JWT.

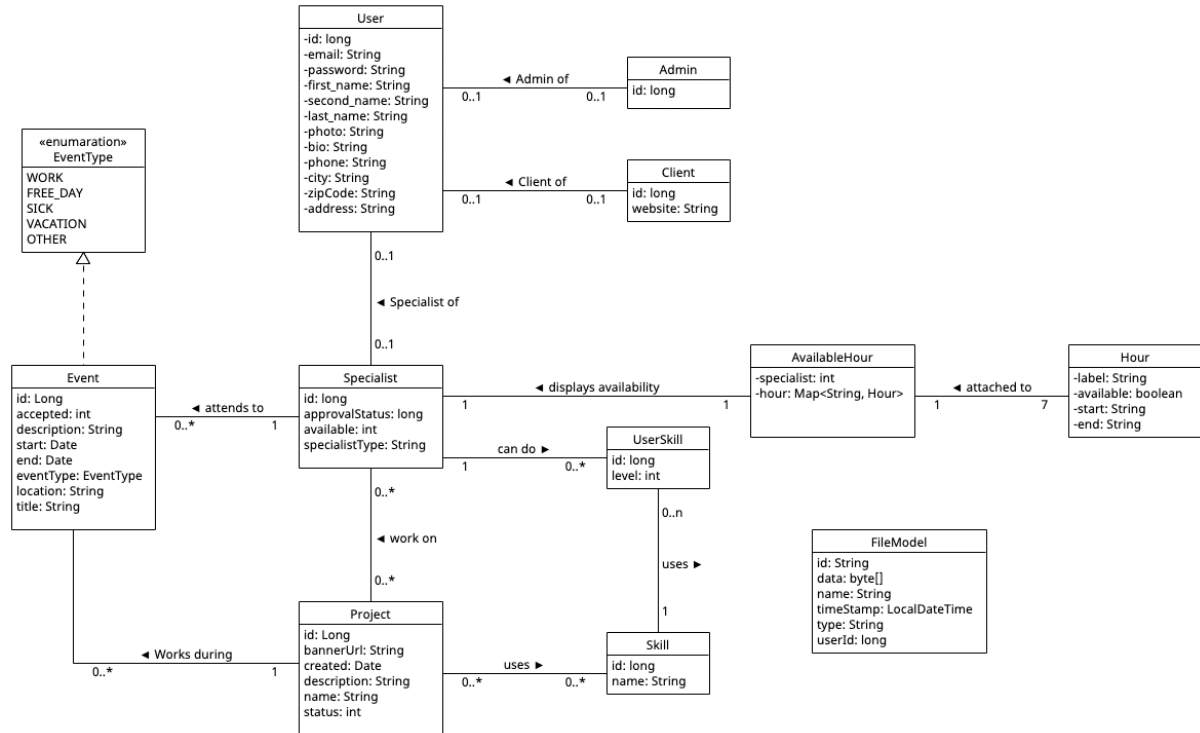
While we did research on other alternatives such as multi-factor authentication (MFA) and Single Sign-On (SSO), we ultimately decided not to implement them in this project. MFA would have added additional complexity to the login process and would have added an extra step for the user. While SSO would have required additional development effort and integration with other systems, which would have been beyond the scope of this project.

Additionally, we researched other possible security features such as encryption of sensitive data, and regular security auditing and monitoring. However, based on the specific requirements of the project and the limited resources available, we decided to prioritize the implementation of JWT over these other features.

In conclusion, the decision to implement JWT for authentication was based on the research we conducted and careful consideration of the specific requirements of the project. While we did evaluate alternatives such as MFA and SSO, we ultimately decided that they were not a good fit for this project. Nevertheless, the security of the application is a constant concern, and we will continue to evaluate and implement new security features as needed.

## 7) Deployment Diagram

Deployment Diagram about a proposal of a physical deployment of the production version of your application at infrastructure that is most suitable for a specific (or general) target audience. Justify your choices.



## 8) Analyzing Design Performance

Our design for a hiring platform for Florijn, aimed to create a seamless and user-friendly experience for clients and specialists alike. We put a lot of focus on the user experience and made sure that our platform is intuitive and easy to navigate, with the ultimate goal of streamlining the process of assigning specialists to projects that were submitted by clients.

Despite our efforts, there are still areas in which the platform can be improved. One such area is the way in which specialists are assigned to projects. Currently, the platform allows for the admin to assign specialists to a project, but we believe that we can take this one step further by introducing a filtering system based on the skills and expertise of the specialists. This way, when an admin creates a project that requires a backend specialist, the admin will be able to see a list of recommended specialists that are specifically skilled in that area. This would help to ensure that the most suitable and qualified specialist is assigned to each project, ultimately leading to better results and increased client satisfaction.

Another area for improvement is in the availability and event system. Currently, specialists can enter when they are available to work, and the client or admin of a project can approve or deny worked hours. This allows for transparency and accountability, but we believe that we can improve upon this by implementing a system that limits the number of hours a specialist can work, based on their availability. For example, if a specialist is only available to work 32 hours a week, the system would prevent them from working more than that. But if the specialist wants to work overtime, the specialist will get an email displaying how much overtime he or she did. This would help to ensure that specialists are not overworked, leading to higher job satisfaction and retention.

In conclusion, while our hiring platform has been well received, we recognize that there is still room for improvement. By focusing on the areas of specialist assignment and availability and event system, we believe that we can enhance the functionality and productivity of our platform, while also reducing costs and increasing client satisfaction. Through further research and testing, we are confident that we will be able to take our platform to the next level.