

|  |  |
| --- | --- |
|  | |
| **Job application training**  User Requirements Specifications | |
| **Team no: DDP1**  Chadi Abdelghani-Idrissi, Arthur Birate Kabanza, Bryan Deckers, Charles Kwakye, Daan Michielsen and Aiana Torgoeva | **2nd phase IT Factory** |
|  |
| Academic year 2022-2023  Geel Campus, Kleinhoefstraat 4, BE-2440 Geel |

**Table of contents**

[Introduction 3](#_Toc122684527)

[1. Assignment description 4](#_Toc122684528)

[1.1. Background information 4](#_Toc122684529)

[1.2. Objectives and target groups 5](#_Toc122684530)

[2. Requirements analysis 6](#_Toc122684531)

[2.1. Functional Requirements 6](#_Toc122684532)

[2.2. Use Case Diagram 7](#_Toc122684533)

[2.3. Use Case Descriptions 7](#_Toc122684534)

[2.3.1. Login 7](#_Toc122684535)

[2.3.2. View announcements 10](#_Toc122684536)

[2.3.3. View files 10](#_Toc122684537)

[2.3.4. Download files 11](#_Toc122684538)

[2.3.5. Book time slot 11](#_Toc122684539)

[2.3.6. Update schedule 12](#_Toc122684540)

[2.3.7. Manage announcements 12](#_Toc122684541)

[2.3.8. Manage files 15](#_Toc122684542)

[2.3.9. Set Booking Deadline 18](#_Toc122684543)

[2.3.10. Create initial timeslot 18](#_Toc122684544)

[2.3.11. Manage emails 20](#_Toc122684545)

[2.3.12. Manage interview schedule 22](#_Toc122684546)

[2.3.13. Archive annual data 23](#_Toc122684547)

[2.3.14. View archived data 24](#_Toc122684548)

[2.3.15. Manage accounts 25](#_Toc122684549)

[2.3.16. Manage company details 26](#_Toc122684550)

[2.3.17. Manage job description 27](#_Toc122684551)

[2.3.18. View time slot report 28](#_Toc122684552)

[2.3.19. Download report 28](#_Toc122684553)

[3. Non-Functional Requirements 28](#_Toc122684554)

[4. PRIORITY PER FUNCTIONALITY 30](#_Toc122684555)

Introduction

We, Chadi Abdelghani-Idrissi, Arthur Birate Kabanza, Bryan Deckers, Charles Kwakye, Daan Michielsen and Aiana Torgoeva, have been given the assignment to find a solution for our client Dirk De Peuter.

The job application training for IT students at our institution has been a successful and valuable opportunity for students to gain practical experience and make connections with potential employers. However, the current system for coordinating and managing the training, which relies on Word and Excel documents and email communication, has proven to be time-consuming and inefficient. In this report, we will analyze the current process and propose a web application as a solution to streamline the administration and communication involved in the job application training.

The goal of our project is to create an online job application training scheduler. The application makes it possible for students to seamlessly book a time slot with up to 3 recruiters from several companies. This application should handle use cases such as booking time slots, update schedule, .... For an overview of all the use case descriptions please refer to 2.2.

# Assignment description

In this chapter we are going to discuss the background information of our project: how it works right now, what modifications are needed and who are the target groups that will be affected by the modification.

* 1. Background information

The current process for coordinating the job application training involves the use of Word and Excel documents and email communication to manage and share information with participating companies and students. This includes maintaining a list of participating companies, the number of recruiters representing each company, and the profiles of students that the recruiters are interested in. It also includes tracking which students have booked time slots for interviews with recruiters.

While this process has been effective in the past, it has become increasingly labor-intensive for the coordinators, who are responsible for maintaining and updating the information on various platforms. In addition, the use of multiple platforms and manual communication can lead to errors and delays in information being shared with the relevant parties.

Our client is responsible for the job application training for the third-year students but currently this is a very time-consuming task due to how everything is set up. He must use multiple programs and must do a lot of the tasks himself.

The client requested that the project should tone down the labor intensity, centralize all the data on one platform, simplify the administration work and make everything more accessible for all parties involved. The client has also mentioned one of the goals is to reduce the usage of Microsoft Office products such as Word and Excel. Also, right now there is a problem with time slot booking efficiency and having to input most of the info of the other parties themselves.

The current way of working goes as follows. The coordinators use Word, Excel and Email for communication and to share documents. The companies and students must register through mail, company information and student information are sent through Word and Excel also through email. We are going to improve this by implementing every feature within one web application. This means that the companies, coordinators and students will be able to sign up and find all the information on one platform.

To improve the efficiency and effectiveness of the job application training, we propose the development of a web application to manage and share information with participating companies and students. This application would allow all parties involved to enter and update information for themselves, reducing the workload on the coordinators and improving the accuracy of the information.

The web application could include features such as a database of participating companies, profiles of recruiters and students, and a scheduling system for interviews. It could also include a feedback system for students to receive feedback and tips from recruiters after their interviews.

* 1. Objectives and target groups

The main objective of the proposed web application is to streamline the coordination and management of the job application training for IT students at our institution. This includes reducing the workload on the coordinators, improving the accuracy and timeliness of information shared with participating companies and students, and providing an efficient platform for scheduling.  
  
In addition, the web application could have the following specific objectives:

* To provide a comprehensive and up-to-date platform of participating companies, recruiters, and student profiles.
* To facilitate the scheduling of interviews between students and recruiters.
* To improve communication and collaboration between the coordinators, participating companies, and students.

There are 3 target groups (**students, coordinators, companies**) who will greatly benefit from the application. As for the coordinators, the work regarding the applications will be more systematic because every needed action can be done within the same application. They can manage time slots, consult data about company's time slots and archived data from previous years.

The primary target group for the web application is the coordinators of the job application training, who will be responsible for managing and updating the application.

The secondary target groups are the participating companies and the IT students. The participating companies will be able to enter and update information about their recruiters and profiles of students they are interested in, as well as schedule interviews with students. The IT students will be able to view the database of participating companies, view the profiles of recruiters and the types of students they are interested in, and schedule interviews through the application. They will have a user-friendly experience which allows them to efficiently choose the options, alter them within the deadlines and retrieve needed information. As for the recruiters, they sort of get rid of the man in the middle and can also request, change and access related data.

In addition to the coordinators, participating companies, and IT students, the target group for the web application could also include the companies that are participating in the job application training. These companies would be able to enter and update information about their recruiters and the profiles of students they are interested in, as well as schedule interviews with students through the application.

Including the participating companies as a target group for the web application would further improve communication and coordination between all parties involved in the job application training. It would also provide an efficient platform for the participating companies to connect with potential hires and provide feedback to students after their interviews.

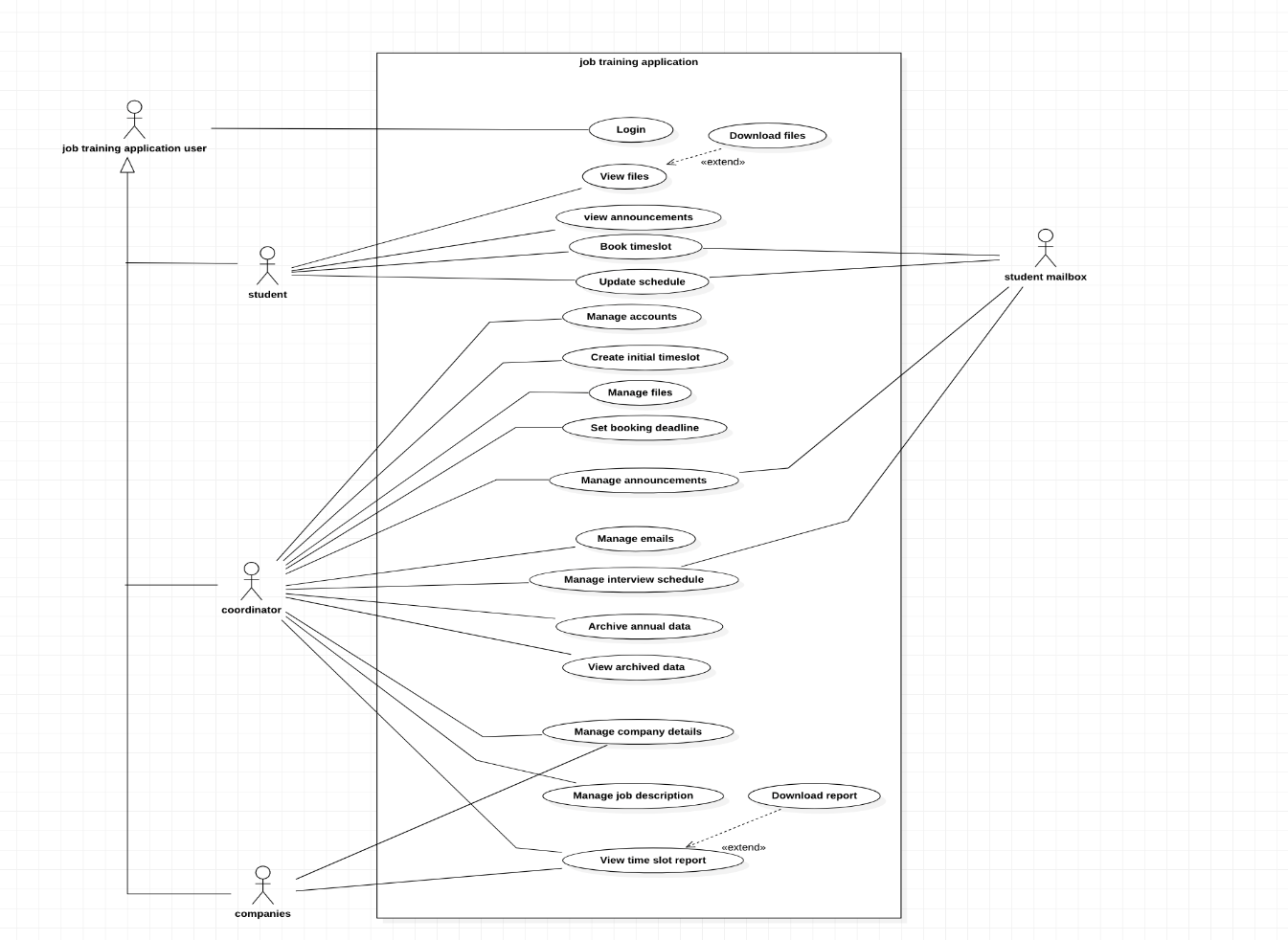
In conclusion, the current system for coordinating and managing the job application training is labor-intensive and can lead to errors and delays in communication. A web application would streamline the administration and communication involved in the training, improving efficiency and accuracy. This solution would benefit all parties involved, including the coordinators, participating companies, and students.

# Requirements analysis

* 1. Functional Requirements

In the Functional requirements we mention all the included use cases and the different behaviors of them, including the descriptions for them. All the use cases of the system are displayed in a use case diagram. The diagram displays the groups that interact with the system and the use cases they can use.

* 1. Use Case Diagram



* 1. Use Case Descriptions
     1. Login

**Functionality:** As a job training application user, I can log in.

**Precondition:** Actor has an account.

**Normal Flow:** The system displays required login details. The actor fills in email and password. The system redirects the actor to the main dashboard.

**Alternative:**

* Incorrect login details: the system displays an error message.
* Forgot password: the system sends an email to the actor which allows the actor to change the password.

1. Screens
2. Login

Graphical user interface, application

Description automatically generated

1. Forgot password

Graphical user interface, text

Description automatically generated with medium confidence

1. Dashboards

* Student

Diagram

Description automatically generated

* Coordinator

Diagram

Description automatically generated

* Company

Graphical user interface, diagram

Description automatically generated

* + 1. View announcements

**Functionality:** As a student, I can view announcements posted by the coordinator.

**Precondition:** Actor must be logged in

**Normal flow:** The system shows all available announcements and attached materials. The actor can select any needed announcement to view the details.

**Screen:**

Graphical user interface, application

Description automatically generated

* + 1. View files

**Functionality:** As a student, I can view files posted by the coordinator.

**Precondition:** Actor must be logged in

**Normal flow:** The system shows all available files. Actor can select any needed file to view the details.

**Screen:**

Graphical user interface

Description automatically generated

* + 1. Download files

**Functionality:** As a student, I can download files posted by the coordinator.

**Precondition:** Actor must be logged in

**Normal flow:** The system shows all available files. Actor can select any needed file to install.

**Screen:**

Graphical user interface

Description automatically generated

* + 1. Book time slot

**Functionality:** As a student, I can book a timeslot.

**Precondition:** Actor must be logged in

**Normal flow:** The system shows all available recruiters with a brief job description, along with a company description. The actor chooses a recruiter. The system displays the available time slots. The actor selects the desired timeslot. The system provides notification and sends a confirmation mail to the actor.

**Screen:**

1. Choose ProfileFunnel chart

   Description automatically generated with low confidence
2. Choose company

Graphical user interface, text, application

Description automatically generated

* + 1. Update schedule

**Functionality:** As a student, I can update my schedule until the deadline set by the coordinator.

**Precondition:** Actor is logged in and has booked a timeslot.

**Normal Flow:** The system displays the registered bookings of the actor. The actor selects the to be changed time slot. The system shows all time slots that are available for the chosen recruiter. The actor chooses the new time slot. The system displays a message of success.

**Alternatives:**

* No time slots available: the system displays a message with contact information of the coordinator.

**Screen:**

Table

Description automatically generated

* + 1. Manage announcements

**Functionality:** As a coordinator I can create and manage announcements.

**Precondition:** Actor is logged in.

**Normal flow:** The system displays all current announcements. Actor can choose to archive old ones, delete them or time them to be posted again later. The system shows confirmation for each action. Actor can also choose to create a new assignment that can be posted at any time. The system sends a confirmation once again.

**Screens:**

1. MainGraphical user interface, text, application, email

   Description automatically generated
2. Publish

Graphical user interface, text, application, email

Description automatically generated

1. Scheduled publish

Graphical user interface, text, application, email

Description automatically generated

1. Unpublish

Graphical user interface, text, application, email

Description automatically generated

1. CreateGraphical user interface, text, application, email

   Description automatically generated
2. DeleteGraphical user interface, text, application, email

   Description automatically generated
3. EditGraphical user interface, text, application

   Description automatically generated
   * 1. Manage files

**Functionality:** As a coordinator I can create and manage files.

**Precondition:** Actor is logged in.

**Normal flow:** The system displays the list of files. Actor can choose to upload new files or delete existing ones. The system shows a confirmation for each action.

**Screen:**

1. MainGraphical user interface, text, application, email

   Description automatically generated
2. UploadGraphical user interface

   Description automatically generated with medium confidence
3. PublishGraphical user interface, text, application

   Description automatically generated with medium confidence
4. Scheduled PublishGraphical user interface, text, application, email

   Description automatically generated
5. DeleteText

   Description automatically generated
6. DownloadText

   Description automatically generated
   * 1. Set Booking Deadline

**Functionality:** As a coordinator, I can set the booking deadline.

**Precondition:** Actor is logged in.

**Normal flow:** The system displays the calendar. The actor sets a date for the deadline. The system asks for confirmation. The actor confirms the deadline.

**Screen:**

Graphical user interface, application, website

Description automatically generated

* + 1. Create initial timeslot

**Functionality:** As a coordinator, I can create an initial timeslot table.

**Precondition:** Actor is logged in.

**Normal flow:** The system displays a timeslot template. The actor sets available times for the future booking. The system asks for confirmation. The actor confirms the selection. The system sends confirmation.

**Screens:**

1. Main

Graphical user interface, application

Description automatically generated

2. Publish Time slots

Table

Description automatically generated

3. Create template

Graphical user interface, table

Description automatically generated

* + 1. Manage emails

**Functionality:** As a coordinator, I can manage emails.

**Precondition:** Actor is logged in.

**Normal flow:** The system displays a list of all active email templates. The actor can view each one and copy or alter them for future use or make a new template as well as delete existing ones. After each action is performed, the system shows a confirmation.

**Screens:**

1. MainGraphical user interface, application

Description automatically generated

2. Create templateGraphical user interface, text, application

Description automatically generated

3. Use templateGraphical user interface, text, application

Description automatically generated

4. Remove template

Graphical user interface, text, application, email

Description automatically generated

* + 1. Manage interview schedule

**Functionality:** As a coordinator, I can manage the interview schedule.  
**Precondition:** Coordinator is logged in, the schedule is approved by company.  
**Normal flow:** The system displays the list of all active companies in the system. The actor chooses the company and adds more or removes time slots to a company’s schedule. The system asks the actor for confirmation. After confirmation by the actor, the system saves the schedule. The actor receives a confirmation email that the changes are recorded. The system is now displaying the saved schedule for the company. The actor proceeds to do the same with the remaining companies.

**Screens:**

1. MainTable

Description automatically generated

2. Create timeslotTable

Description automatically generated

3. Edit timeslotTable

Description automatically generated

* + 1. Archive annual data

**Functionality:** As a coordinator, I can archive annual data.

**Precondition:** The interviews have passed.

**Normal flow:** The system asks for confirmation. The actor confirms to start the archive process by typing archive in the field. The system deactivates the user access for students of this year and displays an archiving successful message and returns to the starting screen.

**Screens:**

Graphical user interface, application

Description automatically generated

* + 1. View archived data

**Functionality:** As a coordinator, I can view the archived data.  
**Precondition:** The actor is logged in.  
**Normal flow:** The system displays a list of all archive years; the actor selects a year he/she wants to consult. The system displays the available data of the selected year. The actor looks at the data and selects close archives when they wish to. The system returns to the starting screen.

**Screens:**

Graphical user interface

Description automatically generated

* + 1. Manage accounts

**Functionality:** As a coordinator, I can manage accounts

**Precondition:** Actor is logged in.

**Normal flow:** The system allows the actor to create new accounts in bulk or one by one for specific cases. The actor creates accounts.

**Alternative:**

* Modify existing accounts: Actor can disable or enable an existing account whenever it’s needed.

**Screens:**

1. MainTable

Description automatically generated

2. New AccountGraphical user interface

Description automatically generated with low confidence

* + 1. Manage company details

**Functionality:** As a company or coordinator, I can fill in the company details.

**Precondition:** Actor is logged in with provided login info. All information is finalized, and the deadline is over.

**Normal flow:** The system displays a set of questions for the company that must be filled out. The actor enters the company name, recruiter name, company description and saves the changes. The system displays a confirmation message and adds the filled in data to the company description.

**Alternative:**

* Modify company details: if the actor chooses, he can also provide further changes to the filled in details such as: recruiter name, company name, company description.

**Screens:**

Graphical user interface, text, application

Description automatically generated

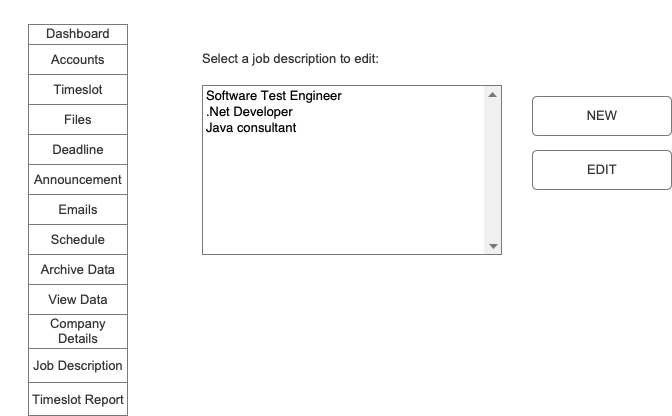
* + 1. Manage job description

**Functionality:** As a coordinator, I can manage a job description.

**Precondition:** Actor is logged in.

**Normal flow:** The system displays the list of all companies. The actor enters the job description information or alters the pre-existing one. The system shows a confirmation.

**Screens:**



* + 1. View time slot report

**Functionality:** As a coordinator, I can view a timeslot report.

**Precondition:** Actor is logged, schedule is confirmed, the deadline for booking the interview is over.

**Normal flow:** The system displays a list of all booked time slots with an option to print or download the table.

**Screens:**

Table

Description automatically generated

* + 1. Download report

**Functionality:** As a coordinator, I can download a report.

**Precondition:** Actor must be logged in

**Normal flow:** The system shows the timeslot report. The actor can download the report.

**Screens:**

# Non-Functional Requirements

**1. Implementation** – The app will be developed using PHP and Laravel. The data shall be stored in a MySQL database, and the schema relational to optimize performance and scalability. Apps should be built using the Model-View-Controller (MVC) design pattern, with the model layer containing all the logic and the view and controller layers containing the user interface.

The app shall have the following features: confirmation and reminder emails for students and companies, and notification emails for students and companies; availability scheduling, allowing coordinators to specify the availability for interviews and students to browse and select available slots; customizable email templates for bulk emailing; file uploading and downloading, allowing coordinators to upload the job description of companies; announcement posting, allowing coordinators to post announcements about job openings or other relevant information; company and job descriptions, allowing companies to provide information about their job opportunities; and a feedback survey, allowing students to provide feedback about their interview experience. The app shall use version control, with all code being tracked in a Git repository.

2. **External Interface** - It should be possible for coordinators to manage interview schedules and retrieve information regarding interviews that have been booked through the application. The app shall send confirmation and reminder emails to students and notification emails to coordinators when an interview is booked or changed. Coordinators shall be able to upload the job description of companies in PDF format, and students shall be able to download these files. The app shall allow coordinators to post announcements about job openings or other relevant information, and it shall support medium integration for posting announcements. Finally, the app will allow companies to provide feedback about their interview experience through a survey.

3. **Performance** - The app should respond to web and API requests in minimum 1 second, normal response time of 2 seconds and maximum response of 4 seconds, handle at least 64 concurrent users, handle at least 50,000 records in the database, and send at least 500 emails per hour without experiencing significant performance degradation.

4. **Quality requirements**

1. Software

The app shall have the following features: confirmation and reminder emails for students and companies, and notification emails for students and companies; availability scheduling, allowing coordinators to specify the availability for interviews and students to browse and select available slots; customizable email templates for bulk emailing; file uploading and downloading, allowing coordinators to upload the job description of companies; announcement posting, allowing coordinators to post announcements about job openings or other relevant information; company and job descriptions, allowing companies to provide information about their job opportunities; and a feedback survey, allowing students to provide feedback about their interview experience.

1. Security

* The app must adhere to industry standards for data confidentiality, integrity, and availability (CIA).
* The app must not reveal sensitive information to third parties.
* The app must not allow unauthorized data access or modification.

1. Hardware

* The application must function well on a range of hardware setups, including desktop and mobile devices.
* The application must be stable enough to deal with unforeseen input or system faults without crashing or losing data.
* To make deployment and upkeep simple, the program must be lightweight and modest in size.

# PRIORITY PER FUNCTIONALITY

The M, S, C, W method is a method for prioritizing requirements in a project. Here's what each letter stands for:

* M: Must have
* S: Should have
* C: Could have
* W: Won't have this time

To use this method, we list all the use cases for your project and then assign each one a priority level based on the above categories. The "must have" requirements are the most important and are necessary for the project to be successful. The "should have" requirements are important but may not be as critical as the "must have" requirements. The "could have" requirements are desirable but not essential, and the "won't have this time" requirements are not included in the current scope of the project.  
  
This method can be helpful for organizing and prioritizing the requirements of a project, as it allows you to clearly identify which requirements are most important and which can be deferred to a later time.

|  |  |
| --- | --- |
| **Use case** | **MoSCoW-priority** |
| Login | M |
| View announcements | S |
| View files | S |
| Download files | S |
| Book time slot | M |
| Update schedule | M |
| Manage announcements | S |
| Manage files | S |
| Set booking deadline | C |
| Create initial timeslot | M |
| Manage emails | C |
| Manage interview schedule | M |
| Archive annual data | S |
| View archived data | S |
| Manage accounts | M |
| Manage company details | C |
| Manage job description | S |
| View time slot report | M |
| Download report | S |