Software Engineering and Project

Software Requirements Specification

PG5

Xuankai Feng

Junjie Zhao

Sharon Khate Damaso

Sai Dheeraj Reddy Pallavolu

Kalpana Parvathaneni

Ying Wang

Yuedong Wang

Lantian Cai

1.0

26/8/2019

Table of Contents

[Version History 1](#_Toc17651773)

[Introduction 2](#_Toc17651774)

[Scope 2](#_Toc17651775)

[Intended audience*.* 2](#_Toc17651776)

[References 2](#_Toc17651777)

[Glossary 2](#_Toc17651778)

[User Requirements 3](#_Toc17651779)

[Use Cases 8](#_Toc17651780)

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Changer | Description |
| 20/8/2019 | 0.1 | Junjie Zhao | Finalized section Introduction. Waiting for final document review |
| 21/8/2019 | 0.2 | Xuankai Feng | Finalized use cases of candidates |
| 23/8/2019 | 0.3 | Xuankai Feng | Finalized use cases of administrators |
| 24/8/2019 | 0.4 | Junjie Zhao | Finalized section User requirements |
| 25/8/2019 | 0.5 | Xuankai Feng | Merged three sections into a smooth report, organized the structure |
| 26/8/2019 | 0.6 | Xuankai Feng | Modify and add use cases |
| 26/8/2019 | 1.0 | Xuankai Feng | Add flow charts to each use case |
| 15/10/2019 | 1.1 | Junjie Zhao | Modify the report base on supervisor’ feedback, add diagrams |

# Introduction

Scope

This document primarily introduces the software requirements from clients of the Hire Me Coder. The main purpose of this project is to develop a system which can permit candidates to get questions and upload answers and allow administrators to manage questions and view answers. The system is a web-based application and obtained for both administrators and candidates.

Intended audience*.*

This document can be read by all related stakeholders.

All development team members are intended audiences. This SRS can be a guide for all members to comply with the management principles and follow the schedule. Moreover, both the scope part and the process part are helpful for developers to ensure all functions that they build can meet the requirements and expectation of the client to a great extent.

Clients from Map Tek are audiences of this report as well. Clients will be interested in all sections of this document since they need to understand our management plan that can help the project get success. Therefore, they can supervise the requirements that the team develop.

References

Software Engineering and project- software plan and management plan template. (2019).

The interview with clients of project specification. (2019).

Tutorialspoint.com. (2019). Scrum - Framework. [online] Available at: https://www.tutorialspoint.com/scrum/scrum\_framework [Accessed 20 Aug. 2019].

Glossary

|  |  |
| --- | --- |
| **Agile Methodology** | A software development approach that promotes continues iteration of development and testing throughout the software development life cycle |
| **GitHub** | Software Development version control |
| **MapTek** | A leading provider of innovative software, hardware and services for the mining industry |
| **Scrum** | An agile process framework |
| **Slack** | Communication Tool |
| **SPMP** | Software Project Management Plan |
| **Trello** | Web-based list-making application |
| **QA** | Quality Assurance |

# User Requirements

**2.1 R001: Login System**

**Description**

The login system allows administrators and candidates to login in the system. This system consists of admin mode and candidate mode. The admin mode allows users to login as an administrator and access to some unique functions for the administrator. And the main administrator can activate and deactivate other administrators. The candidate mode allows candidates to update answers they coded.

**Rationale**

The login system will be used by both administrators and candidates to test the code ability of candidates to ensure all assignments are allocated to suitable candidates. Also, all results of assignments can be uploaded to correct places by candidates.

**Acceptance criteria**

The system can allow users to register and login in the bug-free environment. The system can work smoothly and effectively in most time. This system is the foundation of the whole project. Therefore, this system should be implemented in the early stage of the project period. This function can work with the whole system and share the same database. Safety will be introduced in the safety requirement part.

Only permitted users can access to the system. Therefore, our login system requires valid accounts to allow users to get into our entire system. The administrator accounts can be added by the main administrator. And the candidate accounts should be sent by the administrator via our send email function.

**Source**

Sourced from the first and second client meeting

**Priority**

High

**2.2 R002: Send Email Function**

**Description**

The send email function should be integrated into the admin mode of the login system. The function can only be accessed by administrators, which can send emails to candidates. And this email will include the account that the candidate needed to login to our system. This function can send emails to multiple addresses and preview the email content. Also, since every test has a due time, the administrator can choose the due time before the email is sent. Finally, the administrator can select any tests has been already created in our database to sent to intended candidates.

**Rationale**

The send email function can be used by administrators to send accounts to the candidates. And this function should be the critical part of this project since administrators need to invite intended candidates to finish code tests.

**Acceptance criteria**

The function can automatically generate the unique account for every email address, to be specific, the username should be the email address, and the password should be generated automatically. Candidates can update their passwords in our system.

**Source**

Sourced from the first and second client meeting

**Priority**

High

**2.3 R003 Manage tests**

**Description**

This function is only for administrators. All administrators can manage the assignment database and arrange tests to the intended candidate. Also, administrators can create and delete questions from the database. The administrator needs to arrange up to four questions to create a test. All saved tests and questions can be accessed by administrators.

**Rationale**

This function can be used to help administrators to manage and arrange tests for candidates.

**Acceptance Criteria**

This function would not be touched by candidates; all tests and questions would be saved in our database to ensure the confidentiality of tests. For example, in candidate mode, this function will be hidden by the system.

**Source**

Sourced from client meetings

**Priority**

High

**2.4 R004 View assignment function**

**Description**

This function is for candidates. After the intended receive the invited link, they can click it and view the assignment they need to finish. And this function consists of a time counter; every candidate can view the assignment before the time out.

**Rationale**

This function should be an important component of this project since all candidates need to access tests.

**Acceptance criteria**

This function can work with the whole system. Safety will be introduced in the safety requirement part. And after the due day, the candidate cannot access to the assignment any more since the system will close the page automatically.

**Source**

Sourced from the first and second client meeting

**Priority**

High

**2.5 R005 Upload code file**

**Description**

This function allows candidates to upload the work they have done. After they received an invited link, they can access to assignments and upload the code they developed via this function. This function should cooperate with the send email function. There are two ways to update the answers. First, candidates can update a pdf file which should contain answers they have done; second, they can use the input boxes to enter their answers directly. An online compiler may be applied. Candidates can enter the answers in the editor with embedded complier.

**Rationale**

This function should be a critical part of this project since candidates need to submit their works to let interviewer review their code ability. This function can work with the whole system and share the same database. Safety will be introduced in the safety requirement part.

**Acceptance criteria**

The upload code file function will accept the code file from the intended candidate. Safety will be introduced in the safety requirement part.

**Source**

Sourced from the first and second client meeting

**Priority**

High

**2.6 R006 View submission and provide feedback function**

**Description**

This view submission function should be a part of the login system. Administrators can view the code uploaded by the candidate. The function will provide the upload time and completion time of the assignment. Also, administrators can provide feedback to candidates by this function. The feedback can be commented in the code. When the view stage finish, the system will send result to the targeted candidate.

**Rationale**

This function can be used by administrators to check the quality of the code manually. This function helps the administrator to review the code ability of candidates which improve the efficiency of the process.

**Acceptance criteria**

This function can show the code by candidates in a hierarchical way, which means users can be easy to view the work. Also, the feedback can be sent to candidates with the result of the test in an email. This function can work with the whole system and share the same database. Safety will be introduced in the safety requirement part.

**Source**

Sourced from the first and second client meeting

**Priority**

High

**2.7 R007 Manage candidates’ profile**

**Description**

The candidates can manage their profiles in the system. The profile should include their names, contact numbers, and passwords. Every new candidate may update his/her password when he/she first-time logins to our system.

**Rationale**

The manage candidates profile function help candidates to review all their details.

**Acceptance criteria**

This function can work with the whole system and share the same database. Safety will be introduced in the safety requirement part.

**Source**

Sourced from the first and second client meeting

**Priority**

Medium

**2.8 R101 Safety**

**Description**

The safety requirements ensure this system can keep all users information confidentiality. In this project, user information will keep in the firebase. And the only approved person can view and read the users information.

**Rationale**

Our system will focus on keeping information safe. And we need to pay attention to individual information privacy and avoid leak information.

**Acceptance criteria**

Our system will need to pass a basic cybersecurity test including:

* Vulnerability scanners
* Penetration testing
* Program update checks

And these tests may also include in our development stage.

**Source**

Sourced from the first and second client meeting

**Priority**

High

# Use Cases

To be specific to our functions, we create use cases to help users to understand what the function will help. The below diagram shows the relationship between functions and use cases. The blue boxes are use cases, and the yellow boxes are functions. As we can see, every use case can be allotted to specific function.

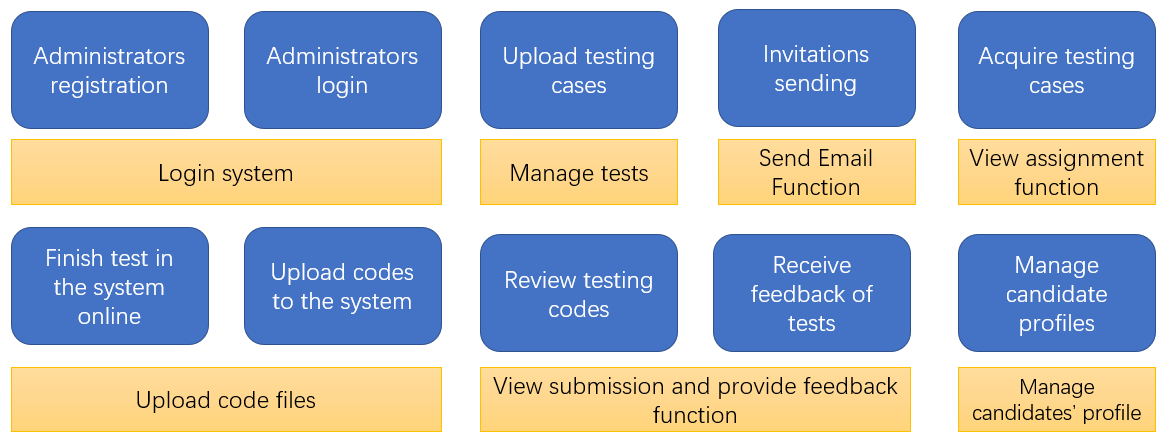


Figure 1. relationship of functions and use cases

**Use Case 001 –** Administratorsregistration

**Brief Description**

Maptek staff use their Maptek email address to register as administrators.

**Actors**

Maptek staff who are responsible for managing or viewing the test in the system.

**Preconditions**

This use case assumes that Maptek staff have their company emails.

**Basic Flow**

1. The Maptek staff visit the system with website browsers
2. The Maptek staff register in the system with their company email addresses
3. The Maptek staff will receive an activation email from the system
4. The Maptek staff finishes activation and fill in basic information and create a password for the system
5. The Maptek staff successfully finish the registration

**Alternate Flows**

* The Maptek staff returns to the homepage or close the website without saving any information

**Exception Flows**

* The Maptek staff input the wrong email address
* The Maptek staff did not receive the activation email
* The Maptek staff insert invalid password

**Post Conditions**

The Maptek staff must receive an email to confirm his registration.

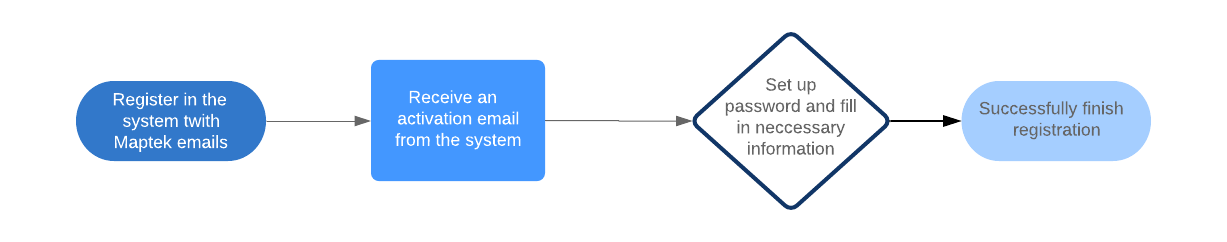


Figure 2. Administrators registration

**Use Case 002 –** Administratorslogin

**Brief Description**

Maptek staff use their registered administrator accounts to log in as administrators.

**Actors**

Maptek staff who are responsible for managing or viewing the test in the system.

**Preconditions**

This use case assumes that Maptek staff had already successfully registered in the system.

**Basic Flow**

1. The Maptek staff visits the system with website browsers
2. The Maptek staff fill the username and password
3. The Maptek staff successfully log in the system

**Alternate Flows**

* The Maptek staff returns to the homepage or close the website with [logging](javascript:;) status

**Exception Flows**

* The Maptek staff forgets his username or password.
* The Maptek staff enter incorrect login details.

**Post Conditions**

The Maptek staff will log in the system as an administrator to manage it.

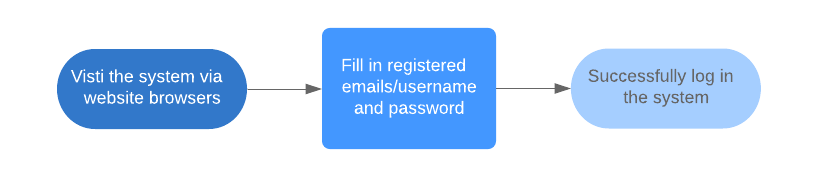


Figure 3. Administrators login

**Use Case 003 –** Upload testing cases

**Brief Description**

Administrators from MapTek who manage the tests use the administrator accounts to upload and create testing cases.

**Actors**

The administrator is anyone who wishes to test the coding ability of candidates by conducting a coding test in their selective programming languages.

**Preconditions**

This use case assumes the administrator had designed appropriate testing cases for candidates in various programming languages.

**Basic Flow**

1. The MapTek staff log in the system with administrator account and password
2. The administrator uploads testing cases in the system backstage
3. The administrator publishes the description and requirement of the test

**Alternate Flows**

* The administrator returns to the previous screen without losing their upload files
* The administrator returns to the previous screen without losing their written description
* A login failure due to network errors

**Exception Flows**

* The system crashes during uploading testing cases
* The uploaded testing cases missing or are incomplete

**Post Conditions**

The administrator must have a local backup of the uploaded testing cases and send invitations to candidates.

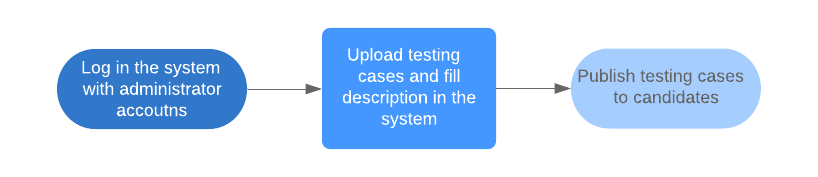


Figure 4. Testing cases upload

**Use Case 004 –** Invitations sending

**Brief Description**

Maptek staff use their registered administrator accounts to send invitations to intended candidates through emails.

**Actors**

Maptek staff who are responsible for inviting candidates to finish tests in the system.

**Preconditions**

This use case assumes that Maptek staff had already successfully registered and logged in the system.

**Basic Flow**

1. The Maptek staff logs in the system as an administrator
2. The Maptek staff send an invitation to the candidates for tests.

**Alternate Flows**

* The Maptek staff returns to the homepage or close the website without sending invitations.

**Exception Flows**

* The Maptek staff sends invitations to incorrect email addresses.

**Post Conditions**

The invited candidates will log in the system and finish tests.

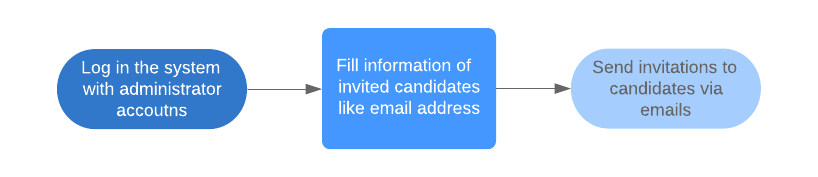


Figure 5. Sending invitations

**Use Case 005** – Manage profiles

**Brief Description**

Candidates can modify the account name, contact number and password.

**Actors**

Candidates can update their information after they login to our system.

**Precondition**

This use case assumes that candidates can login to the system.

**Basic flow**

1. The candidate signs in with the username and password allocated by Maptek
2. The candidate login to the system
3. The candidate can update the password, account name and contact number

**Alternate actions**

N/A

**Exception flow**

* Candidates enter an invalid password when they update the password

**Post condition**

N/A

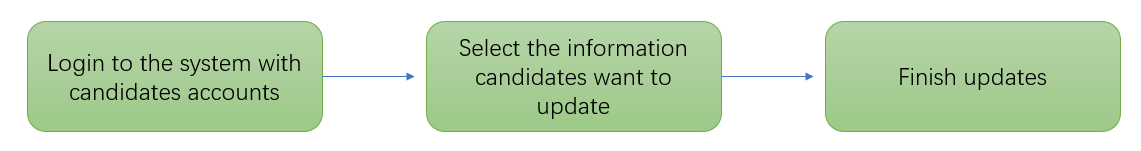


Figure 6. Manage profile use case

**Use Case 006 –** Acquire testing cases

**Brief Description**

Invited candidates who wish to get access to the testing cases provided in the system by MapTek.

**Actors**

The candidate is anyone who wishes to complete an online test via uploading their codes finished in their computers or writing codes in the system online.

**Preconditions**

This use case assumes that the candidate can view the testing cases and submit his codes in the system.

**Basic Flow**

1. The candidate signs in with the username and password allocated by Maptek
2. The candidate selects the programing language in the system
3. The candidate views the testing cases and detailed requirements
4. The candidate finishes coding online or on his device and saves it as a file
5. The candidate uploads or submits his codes in the system

**Alternate Flows**

* The candidate returns to the homepage to reselect programming languages after viewing the details of a testing case
* The candidate did not upload or submit any codes after the due date

**Exception Flows**

* The descriptions of testing cases are missing or incomplete

**Post Conditions**

The candidate must receive an email to indicate the due date and following steps.

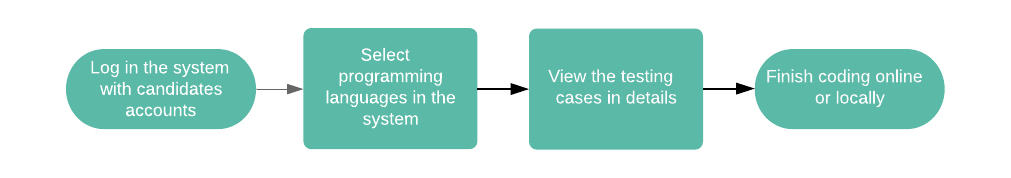


Figure 7. Acquire testing cases

**Use Case 007 –** Finish test in the system online

**Brief Description**

Invited candidates who wish to write codes in a specific programming language to complete the test online.

**Actors**

The candidate is anyone who wished to write his codes online, does not create any local files, does not take advantage of local integrated development environments or programming software.

**Preconditions**

This use case assumes the candidate can finish the testing case via the online editor of the system.

**Basic Flow**

1. The candidate signs in with the username and password allocated by Maptek
2. The candidate read the description and requirement of the testing case
3. The candidate writes codes to finish the test via the online editor
4. The candidate submits his codes to the system for MapTek review
5. The candidate receives a message to inform the following steps and wait for further response

**Alternate Flows**

* The candidate returns to the previous page without losing his codes
* The candidate paused his coding for a while without losing his codes
* The candidate closes the page without losing his codes
* The candidate turns off his computer without losing his codes

**Exception Flows**

* The system crashes by chance during online coding
* The system loses the candidate’s codes after he submits

**Post Conditions**

The candidates must have a successful submission at the end, to ensure the follow-up recruitment process. And wait for the feedback from Maptek.

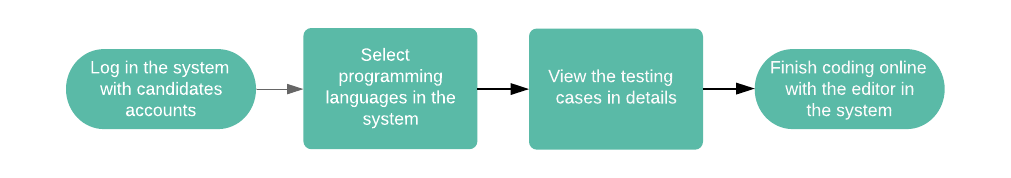


Figure 8. Finish test in the system online

**Use Case 008 –** Upload codes to the system

**Brief Description**

Invited candidates who wish to finish their tests may prefer to use the local programming software to write codes.

**Actors**

The candidate is anyone who wishes to write codes locally, do not write codes online, do not wish to finish coding in the system with the embedded editor.

**Preconditions**

This use case assumes the candidate can utilise local integrated development environment to finish tests on time.

**Basic Flow**

1. The candidate signs in with the username and password allocated by Maptek
2. The candidate read the description and requirement of the testing case
3. The candidate writes codes to finish the test via the local integrated development environment and programming software
4. The candidate uploads his local codes to the system for MapTek review
5. The candidate receives a message to inform the following steps and wait for further response

**Alternate Flows**

* The candidate returns to the programming languages selection after uploading codes.

**Exception Flows**

* The system becomes jammed during uploading local codes
* The system loses the archive of uploading codes

**Post Conditions**

The candidates must have a successful submission at the end, to ensure the follow-up recruitment process. And wait for the feedback from Maptek.

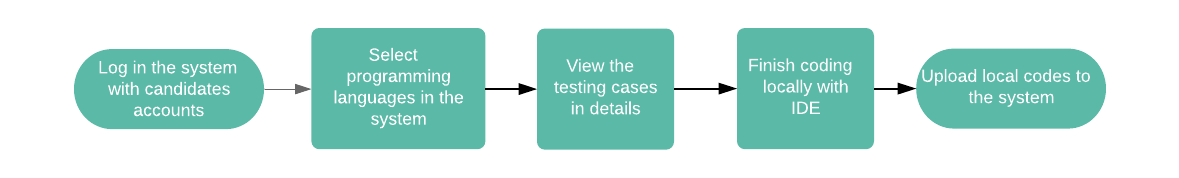


Figure 9. Upload codes to the system

**Use Case 009 –** Review testing codes

**Brief Description**

Administrators from MapTek who manage the tests use the administrator accounts to download and review testing codes submitted by candidates

**Actors**

The administrator is anyone who wishes to test the coding ability of candidates by conducting a coding test in their selective programming languages.

**Preconditions**

This use case assumes the administrator had published appropriate testing cases for candidates in various programming languages, and candidates had finished and submitted their testing codes.

**Basic Flow**

1. The MapTek staff log in the system with administrator account and password
2. The administrator downloads testing cases submitted by candidates in the system backstage
3. The administrator compiles testing codes locally
4. The administrator marks the testing codes for all candidates
5. The administrator publishes feedback and results on the system for candidates’ review

**Alternate Flows**

* The administrator returns to the previous screen without losing their written feedback in the system
* The administrator returns to the previous screen without losing their uploaded notifications

**Exception Flows**

* The system crashes during uploading feedback and results
* The system fails in published corresponding feedback to candidates

**Post Conditions**

The administrator must have a local backup of the feedback and results of all candidates, and candidates will receive feedback.

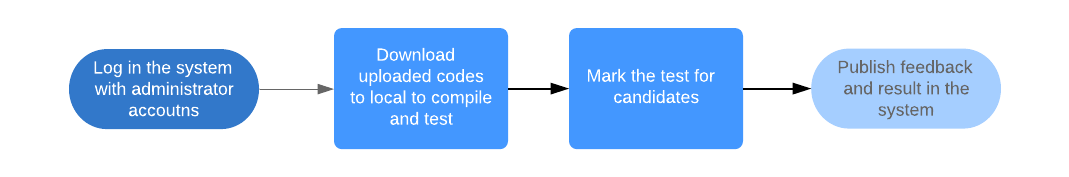


Figure 10. Review testing codes

**Use Case 010 –** Receive feedback of tests

**Brief Description**

Invited candidates who had taken part in the coding test may want to receive feedback and result from MapTek.

**Actors**

The candidate is anyone who wishes to receive feedback for their submitted codes and the result of his application.

**Preconditions**

This use case assumes the candidate can get access to the system and read messages on it.

**Basic Flow**

1. The candidate signs in with the username and password allocated by Maptek
2. The candidate read the feedback of his codes from Maptek
3. The candidate read the result of application and follow-up actions from Maptek

**Alternate Flows**

* The candidate does not get access to the system and view the feedback

**Exception Flows**

* The feedback from MapTek did not appear in the expected time
* The follow-up actions are missing or blurry

**Post Conditions**

The candidate must have a clear understanding of feedback and follow-up actions.

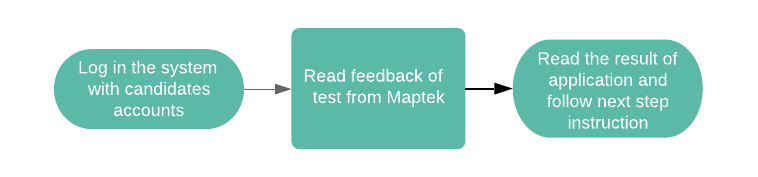


Figure 11. Receive feedback of tests

# Appendix A

Figure 1. Administrators registration

Figure 2. Administrators login

Figure 3. Testing cases upload

Figure 4. Sending invitations

Figure 5. Mange candidate profiles

Figure 6. Acquire testing cases

Figure 7. Finish test in the system online

Figure 8. Upload codes to the system

Figure 9. Review testing codes

Figure 10. Receive feedback of tests