



Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: CSION

Analysis Report

İsmail Yavuzselim Taşçı, Ayça Begüm Taşçıoğlu, Mehmet Sanisoğlu, Nursena Kurubaş, Mehmet Selim Özcan

Supervisor: Prof. Dr. Uğur GÜDÜKBAY

Jury Members: Prof. Dr. Özgür Ulusoy and Prof. Dr. İbrahim Körpeoğlu

Contents

1 Introduction	3
2 Proposed System	3
2.1 Overview	3
2.2 Functional Requirements	4
2.3 Non-functional Requirements	5
2.3.1 Usability	5
2.3.2 Efficiency	5
2.3.3 Response Time	6
2.3.4 Reliability	6
2.3.5 Security	6
2.3.6 Maintainability	6
2.3.7 Scalability	6
2.3.8 Recoverability	6
2.4 Pseudo Requirements	7
2.5 System Models	8
2.5.1 Scenarios	8
2.5.2 Use-Case Model	12
2.5.3 Object and Class Model	13
2.5.4 Dynamic Models	14
2.5.5 User Interface	19
3 Other Analysis Elements	27
3.1 Consideration of Various Factors	27
3.2 Risks and Alternatives	29
3.3 Project Plan	30
3.4 Ensuring Proper Team-Work	33
3.5 Ethics and Professional Responsibilities	33
3.6 New Knowledge and Learning Strategies	33
4 References	35

Analysis Report

Project Short-Name: CSION

1 Introduction

Throughout life, people are bound to experience certain ups and downs, as such is life. In several cases, we come across a type of crossroads, in which we try to find the optimum outcome. However, it is not always easy to take every aspect of the problem into account and make a rational decision. In such cases, the advice of a friend is always welcome as it provides a different view. However, the advice you get is not exactly bias-free let alone optimal for the situation you are in. *Csion* however, relies on your inner self, as well as your current mindset and as such, is a reflection of yourself, giving you advice in an objective, rational and personalized manner.

The following parts of the report provide detailed analysis of the system we propose in part two, with requirements analysis and model specifications via models. The models provided define the system we propose with class, activity and sequence diagrams. Lastly, we have included several mock-ups for the illustration of the user interface. The report follows with the third part where we investigate other analysis elements, in which we explore several risks and factors that affect the project. We also provide the plans we have made for the course of action we want to take in this project. In the last part, we have included a references section to complete our report.

2 Proposed System

In this part, we have examined the system we have come up with for the project and analyze the requirements for the suggested plan.

2.1 Overview

Csion aims to optimize people's decisions about any subject based on their characteristic behaviours within a short amount of time. We are planning to use Myers-Briggs Personality types to have an initial categorization of the users' characters [1]. This test is highly standardized and is esteemed in the field of psychology. There are 16 unique personalities and each person can find their personality type by solving simple and quick test questions. These types have certain borderline character traits that generally conform to the way people behave. We will use the "Crystal Know" API to implement this test and get the results to use inside our application. Crystal Knows will also provide us data of over 6 million individuals who have done this test.

Although it is a deciding factor, personality alone is not the only variable when a person makes a decision. A detailed analysis of the subject that the user will decide, also needed to provide reliable suggestions. Because of that, in our project *Csion*, we will implement a Chatbot which uses Natural Language Processing techniques to give detailed analysis of a sentence or paragraph. We will ask from users to write the problem itself, pros and cons. Then by combining that analysis and the person's general personality, we will give detailed suggestions to our users to help them decide.

Another way of using *Csion* will be through our question-answer section. In that section, users will only be able to get suggestions from predetermined categories such as changing company or changing department etc. After selecting the problem, we will ask predetermined questions to the user to get important information that affect the decision for that problem. After getting answers from users, we will use scientific methods to combine personality of the user and the answers that we get to determine what users

should decide. After this processing phase, we will again show a detailed suggestion to user. To increase usability of this section, questions will only include only yes or no questions and users will answer the questions by using a very convenient way. For yes, users will swipe to left and vice versa. In case of any wrong question, another answer will be the irrelevant and to answer as irrelevant users will swipe to top.

After giving suggestions, we will ask user to enter a deadline for the decision. After deadline day, we will ask user to give feedback to our suggestion whether he chose what we suggested or not and his satisfaction rate about his decision. With using this feedback data and deep learning implementation, we will make our application learn and improve itself to make better and better suggestions for future questions.

2.2 Functional Requirements

This part is about the functions of the application that affects user experience. To see the functionality of the application in more detail, see section 2.5.5.2.

Sign-Up/Log-In Process

When users open the application for the first time after downloading, they will encounter a page with two options: Login and Sign Up. Users have to register to use the application, but if they are already registered, they can just click "Login" button and login by typing the required information and clicking "Login".

To register to the application, users need to click "Sign Up" and fill the boxes that requires certain personal information. After completing the registration form, as the last step of registration, users should solve a personality test. Afterwards, the system will automatically login the user to the application.

Asking for a Decision

When a new session is created, user will see the Main Page and it will have two different tabs: "Choose Category" and "Ask to Chatbot". It is recommended for users to choose Category system if their questions has a match there, but for all other complex questions, they are free to use are Chatbot system.

If users choose the first tab, they will be able to choose between different categories. When they click a category, subcategories of that category will open and users will select the category that is related with their doubts. After selecting the desired question, a deadline should be selected for the decision –as a way of saying "I should decide this until this date". Then the system will start to ask users several questions related to the topic to increase the accuracy of the final decision, but users will always be able to skip to the final decision by clicking the "Skip Questions" button. Users will be able to answer the questions by swiping question boxes to right for "No", left for "Yes", and in case it is needed, up for "Irrelevant". After clicking the skip button or completing all the questions, the final decision that arranged specifically for the user will be presented at the analysis page.

If users choose the Chatbot tab, they will see a form. Users should type their question, and pick a deadline and it is recommended for them to also fill the pros and cons list in order to get more accurate results. After submitting the form, users will see the analysis page.

Profile

Each user will have a profile that having their personal information, personality types that is obtained from tests, a "Former Q/A" section that users will be able to see their former decisions, the advice they took from the application, the decision they made and how content they were with this decision. They will also be able to give feedback to their former questions through this page. Profile page will also include "Additional Tests" section that

provide users other personality tests to solve them and discover other aspects of their personalities.

Giving Feedback

When the selected deadline comes, the application will send the user a notification that is asking what did they decide to do, and after clicking to the notification or opening the application, users should select what they chose to do and how much they are content with their decision from "Former Q/A" section at profile page.

Settings

Users will be able to reach the settings page through a button at navbar that is represented with a cogwheel icon that remains at the main page. Settings page will include account settings, like removing or suspending the account or visual settings like themes, notification settings, an "About Us" section for users to have knowledge about the contributors of the application, a "Help" section to give user further information about application and a "Terms & Conditions" part for the users who might want to read it again. Users will also be able to change their password from here.

2.3 Non-functional Requirements

In this section we have listed primary non-functional requirements for Csion.

2.3.1 Usability

- Application should be understandable for every user. User interface will provide every information that user can access.
- Menus and sub-menus should be usable for every portion of the application. Also, menus should be easy to tap and easy to read since users can have difficulties for reading.
- Application should provide information such as tips and manuals for users when they first login to the system.

2.3.2 Efficiency

- Number of questions should not exceed 10 for categorized problems.
- Questions must cover at least 1 keyword of the personality type in order to decrease the number of questions.
- IBM Watson API customization must cover every keyword for every personality and result of the analyzed text must return at least 90% of the related keywords.
- Analyzed output should be direct and should give only necessary percentages.

2.3.3 Response Time

- Answers of the questions and any inputs of user should be processed less than 1 second.
- Data transfer between Database and server should last less than 1 second for login and signup operations.
- Communications between servers should last less than 1 second.
- Application should create a final analyzed output less than 10 seconds.

2.3.4 Reliability

- Application should provide exact results of personality tests without any errors.
- Result of analyzed text that comes from IBM Watson should return correct emotion, correct keywords, correct overall sentiment, correct concept and correct syntax.
- Application should label each answer of questions correctly and their value as a variable for algorithm. Algorithm must work coherent for same inputs.

2.3.5 Security

- The answers of questions should be accessible only by the user should they decide not to give feedback.
- Outside of the application, the result of personality tests should be only accessible by Crystal Knows since Crystal Knows has right to keep the results of tests.
- Any given personal information such as passwords and social platform information should be encrypted and protected that only user and application developers can access.
- Personal data of the user should not stored if the user deletes it's account

2.3.6 Maintainability

- Application should be fixable for less than an hour in case of any wrong output
- Application should be updatable for future implementations for less than an hour.
- Servers should be checked for any undesired behavior for every day

2.3.7 Scalability

- Application should process more than 100 thousand personal data.
- Application should answer more than 2500 users simultaneously.
- Database should be scalable upto 100 thousand users for the first stage.

2.3.8 Recoverability

- In case of a system shutdown or unexpected failure, application should return a functioning state and remember the questions that user has already answered.
- Database should keep the state of the last safe situation and in case of any problem database should return this state.

2.4 Pseudo Requirements

1. The system will have a server to keep data of its users, natural language processor to understand given question and answers, and a predictor to give an answer to the given question according to user's data and previous answers of the users' with similar personality types.
2. Program will be able to used in both Android and IOS devices.
3. Github will be used as the project repository for source codes and reports. Developers will commit to repository simultaneously.
4. Object-Oriented Programming will be used for software design.
5. External libraries will be used for natural language processor, deep learning process and prediction.
6. Appium will be used for application automation.
7. Crystal Knows API will be used to take personality test and integrating it to program.
8. Price of Crystal Knows API will be paid by developers, monthly.
9. Additional tests will be provided to users, information about the user will be expanded with these additional tests.
10. IBM Watson Chatbot will be used the Natural Language Processor of the program.
11. IBM Watson API will be integrated to the program for "Ask to Chatbot" part.
12. After the CSION will be released in Google Play and App Store, Textrazor API will be added to improve Natural Language Processor of the program.
13. Program will be developed in cross platform environment.
14. Server of the program will be implemented with Node.js.
15. MariaDB will be used as the database of the program.
16. User provides feedback for program's analysis and predictions. After user provides feedback, deep learning algorithm will be built to give more accurate predictions.
17. User data will be encrypted and will not be shared with Third-Party Applications.
18. Modules will be tested internally, then they will be implemented. The modules will be in a standalone working state, where each will be ready to operate within themselves. Then, they will be combined together to enable the workflow.
19. System may be extended in the future for multiple purposes such as hiring system or may be extended with additional features such as working with real life tutors and matching customers with them regarding to customer's problems and personality.

2.5 System Models

This section covers the possible scenarios the users can go through and provides the relevant Class, Sequence and Activity diagrams for these cases.

2.5.1 Scenarios

The following are excerpts from the set of possible courses the users can take by using the application.

Scenario 1: Sign Up

Actors: New user

Conditions: New user opens the app

Exit Conditions: User is navigated to their Main Page.

Flow of Events:

1. User clicks "Sign Up" button on the first page of the application.
2. The registration page is opened.
3. User specifies the account's username, name, surname, gender, birthday, email and password.
4. User is redirected to the Personality Test Screen.
5. User clicks the "Take the Test" button and fills personality test.
5. System detects personality type of user.
6. System creates an account for the user stores personality test data with user's account information and adds account to the database.

Scenario 2: Log In

Actors: Registered User

Entry Conditions: User opens the app.

Exit Conditions: User closes the application.

Flow of Events:

1. User clicks "Log In" button on the first page of the application.
2. User types their username and password.
3. User clicks "Log in" button on Log In Screen.
4. System matches given username and password pair with the pair in database.
5. The home page is opened.

Scenario 3: View Personality Test Results

Actors: User

Entry Conditions: User is in their Profile Page.

Exit Conditions: User clicks another tab on Profile Page or clicks Home Page Button.

Flow of Events:

1. User clicks Personality Tab on their profile page.
2. System provides the character type, previous test results and analytics about personality to user.
3. User views previous character type, test results and analytics.
4. User clicks another tab on Profile Page or clicks Home Page Button.

Scenario 4: Retake the Personality Test

Actors: User

Entry Conditions: User is in their Profile Page, Personality Tab.

Exit Conditions: User completes personality test again or discard changes.

Flow of Events:

1. User clicks "Take the test again!" button on their Personality Tab in Profile Page.
2. Program warns the user about that if they retake personality test again, previous activity will disappear.
3. User confirms to retake the test and deleting previous account data.
4. User completes the retaken test and completes.
5. System updates the account data of the user.
6. User is redirected to their homepage.

Scenario 5: Choose Category for a Decision

Actors: User

Entry Conditions: User is in their home page.

Exit Conditions: User clicks "Home Page" button in the result page and is redirected to their homepage.

Flow of Events:

1. User clicks "Choose Category" button on Main Page.
2. User is redirected to Category Screen.
3. User selects a category.
4. User selects a subject about that specific category and presses "Next" button.
5. System displays questions from its question pool with related category and subject.
6. User swipes left or right for answering questions.

7. Program analyzes user's answers and picks related questions from question pool until the program reaches the nearest answer with respect to the user's personality test results, and answers.
8. Program displays the answer to the user's question, shows the reasons of its answer regarding personality test result.
9. User clicks "Home Page" button on result page and is redirected to their home page.

Scenario 6: Ask to Chatbot

Actors: User

Entry Conditions: User is in Main Page.

Exit Conditions: User clicks "Home Page" button in the Analysis Screen and is redirected to their homepage.

Flow of Events:

1. User clicks "Ask to Chatbot" tab on the Main Page.
2. User types the complex question to "Question" text field.
3. User types pros in "Pros" text field.
4. User types cons in "Cons" text field.
5. User picks a deadline
4. User types their question.
5. Program detects keywords and understands problem, its beneficial and counter effects with its Natural Language Processor.
6. Program generates an answer to user's problem with respect to the personality test result of the user, problem and pros and cons of the problem.
7. User clicks "Exit" button on the result page and is redirected to their home page.

Scenario 7: View Previous Questions and Answers

Actors: User

Entry Conditions: User is in their Profile Page.

Exit Conditions: User clicks another tab in Profile Page.

Flow of Events:

1. User clicks "Former Q/A" tab on their Profile Page.
2. System provides the previous questions and answers of user with application's answers and analysis.
3. User views previous questions, answers and analysis.
4. User clicks another tab in Profile Page.

Scenario 8: Take Additional Test

Actors: User

Entry Conditions: User is in their Profile Page.

Exit Conditions: User clicks another tab in Profile Page.

Flow of Events:

1. User clicks "Additional Tests" tab on their Profile Page.
2. User picks an additional personality test from provided additional tests.
3. User clicks "Take the Test!" button.
4. User is redirected to additional test.
5. User completes additional test.
6. System collects additional personality test results and stores them in database with the account information.
7. User clicks another tab in Profile Page.

2.5.2 Use-Case Model

The use case diagram that illustrates the possible actions a user can take is provided below.

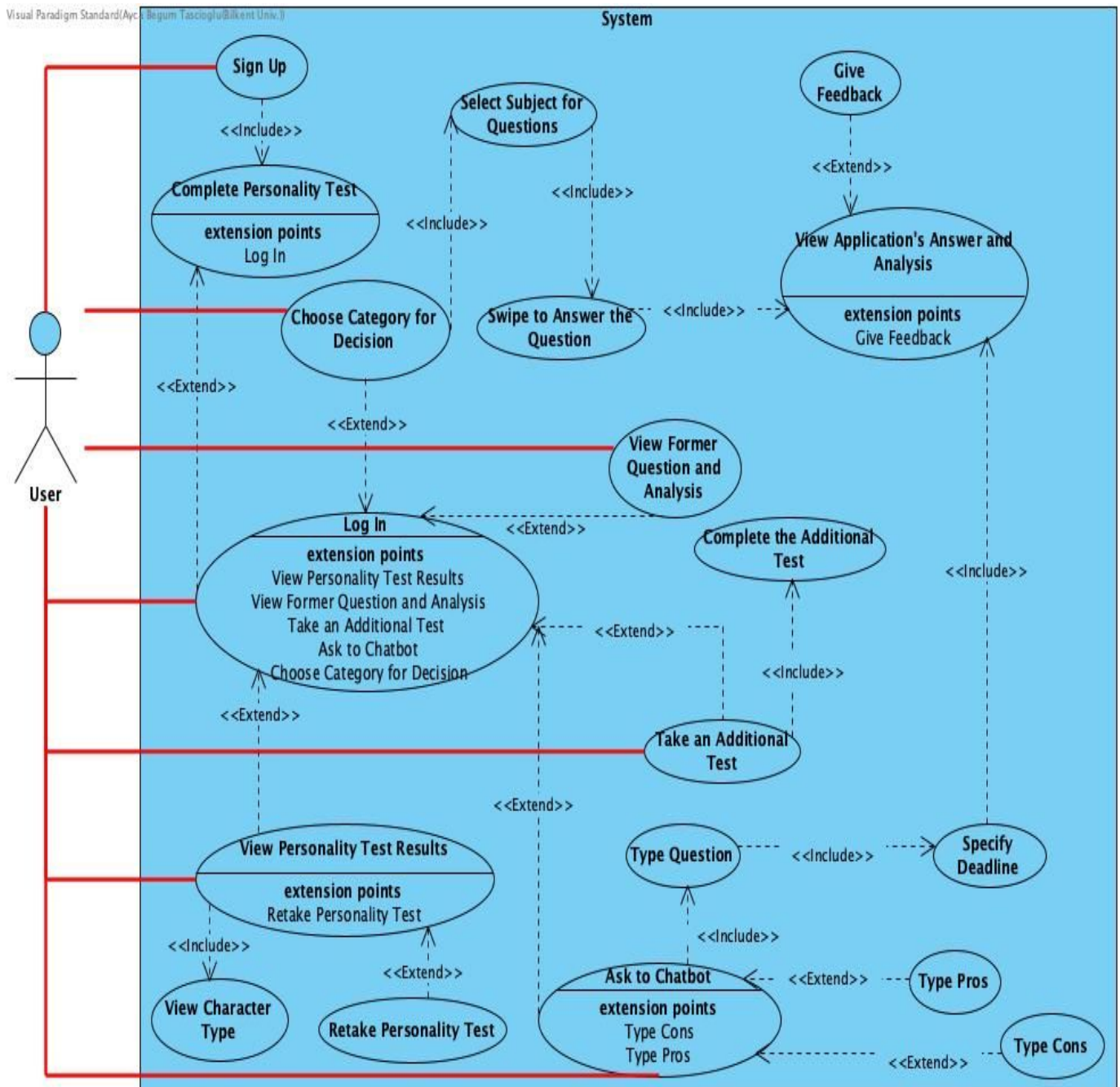


Figure 1. The use case diagram for Csion

2.5.3 Object and Class Model

The following figure is the general design for the class model. We are using the client-server approach for our application. We have put the main logic in the server side and UI implementation in the client side. Also we are using additional servers like CrystalKnows and IBM Cloud Watson server.

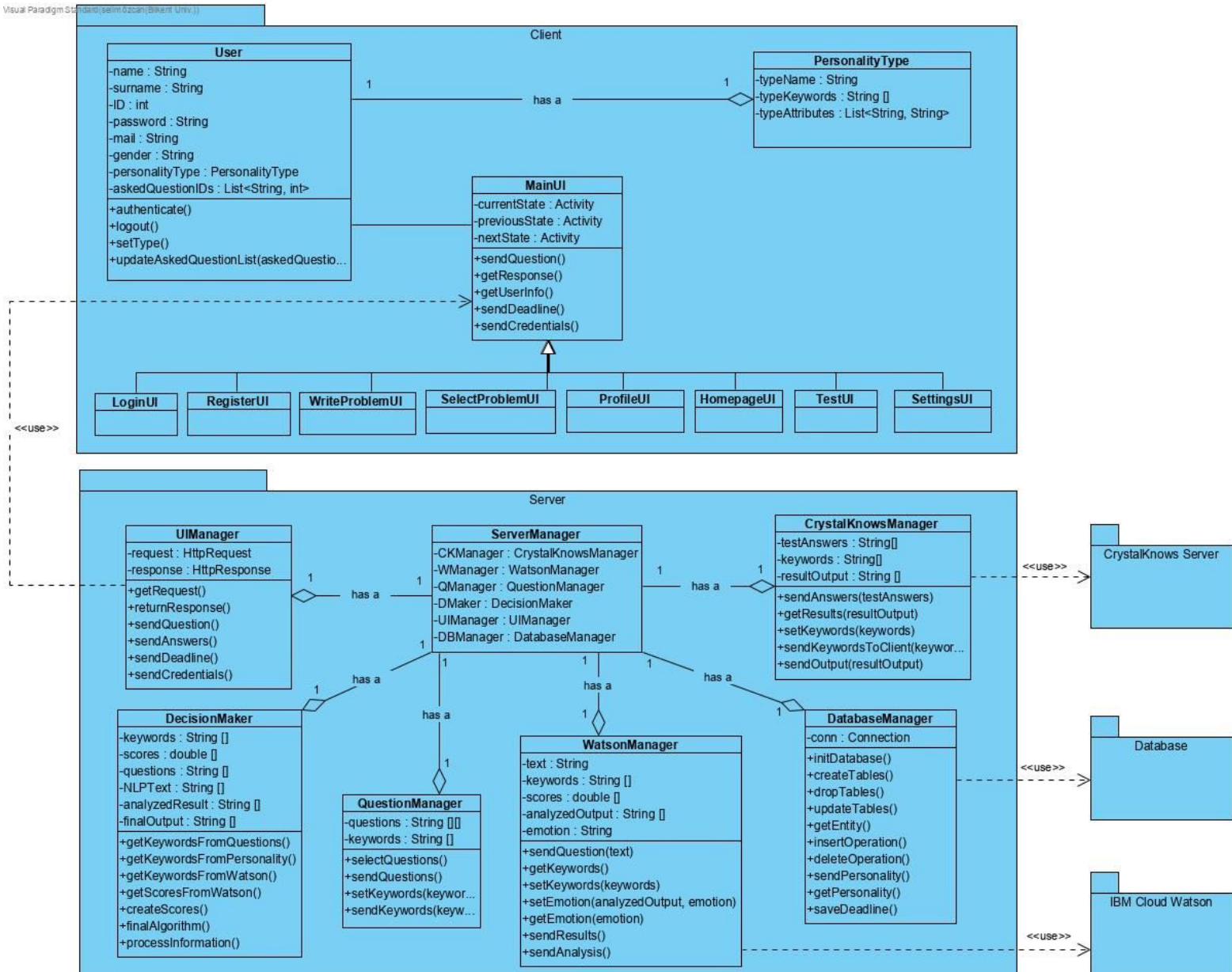


Figure 2: Object and Class Diagram for Csion

2.5.4 Dynamic Models

In this section we have created class, activity and sequence diagrams for the project. The diagrams can be found in their respective sections.

2.5.4.1 Sequence Diagrams

The following are the sequence diagrams for Sign up, Login and Taking Psychology Test, Decision Making activities respectively.

For better quality, you can click these links:

For Sign Up: <https://i.hizliresim.com/Z5BNn0.jpg>

For Login: <https://i.hizliresim.com/P7W3Gb.jpg>

For Decision Making: <https://i.hizliresim.com/gP0lbQ.jpg>

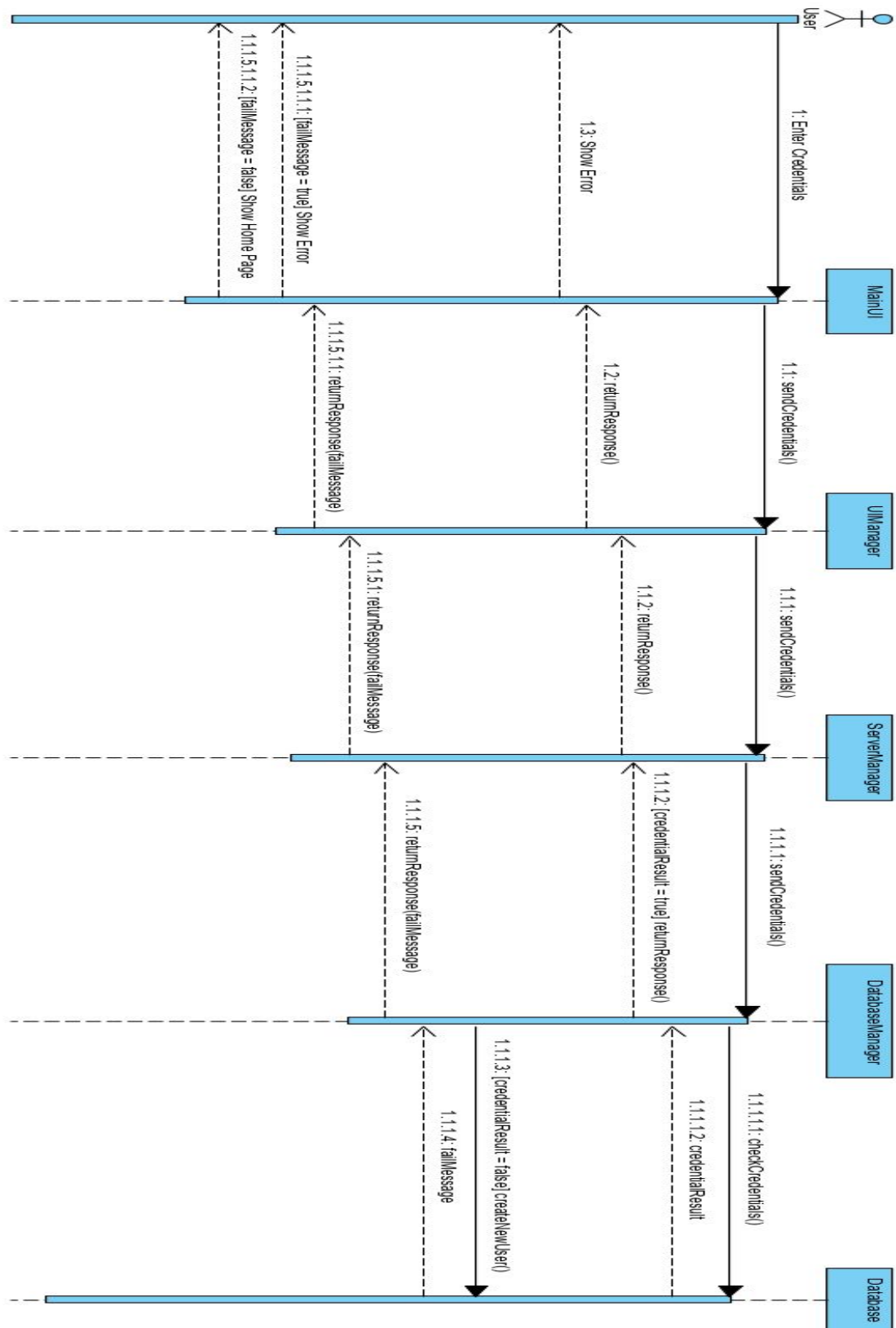


Figure 3. Sequence Diagram for Sign up Activity

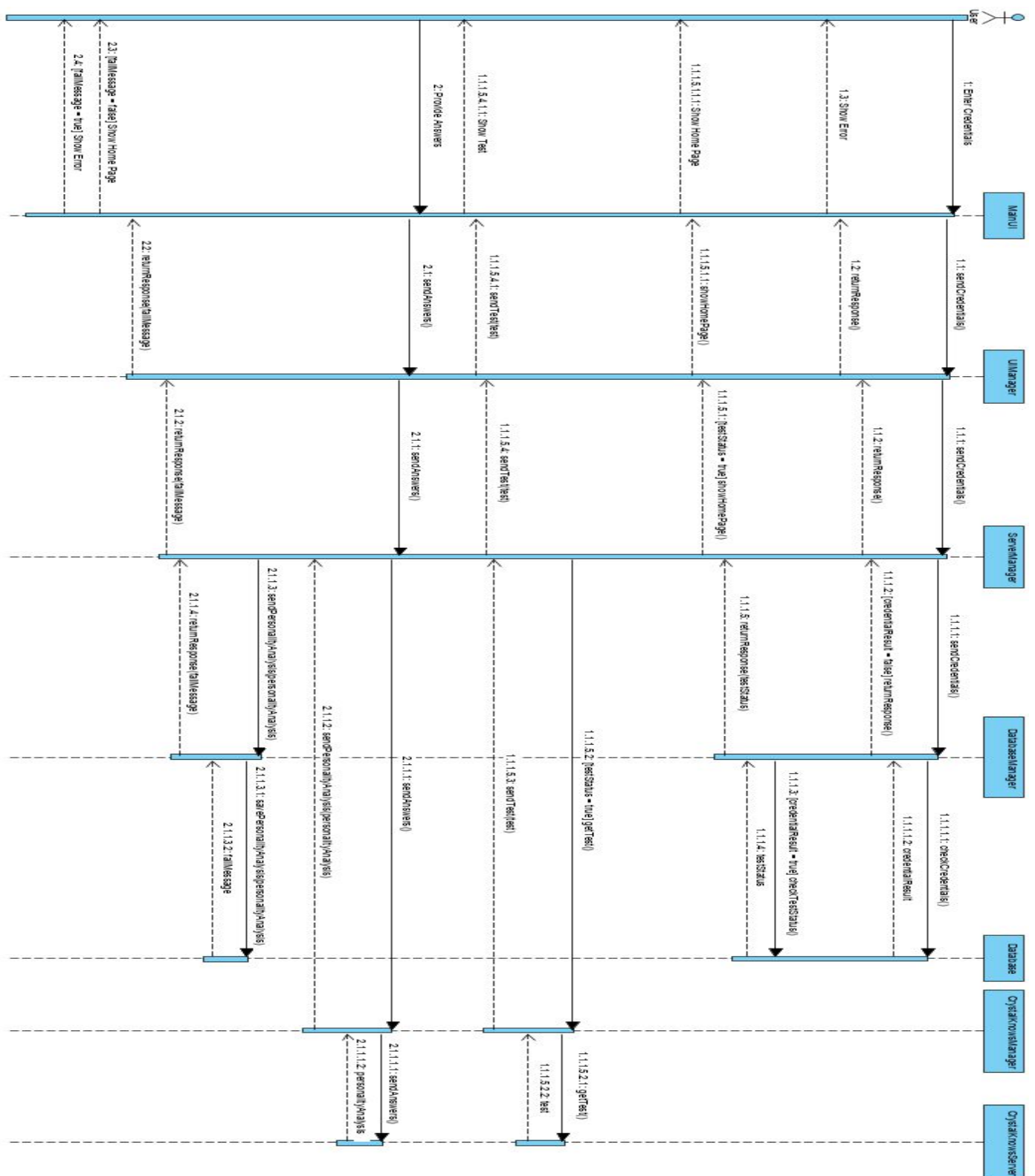


Figure 4. Sequence Diagram for Login and Taking Psychology Test Activity

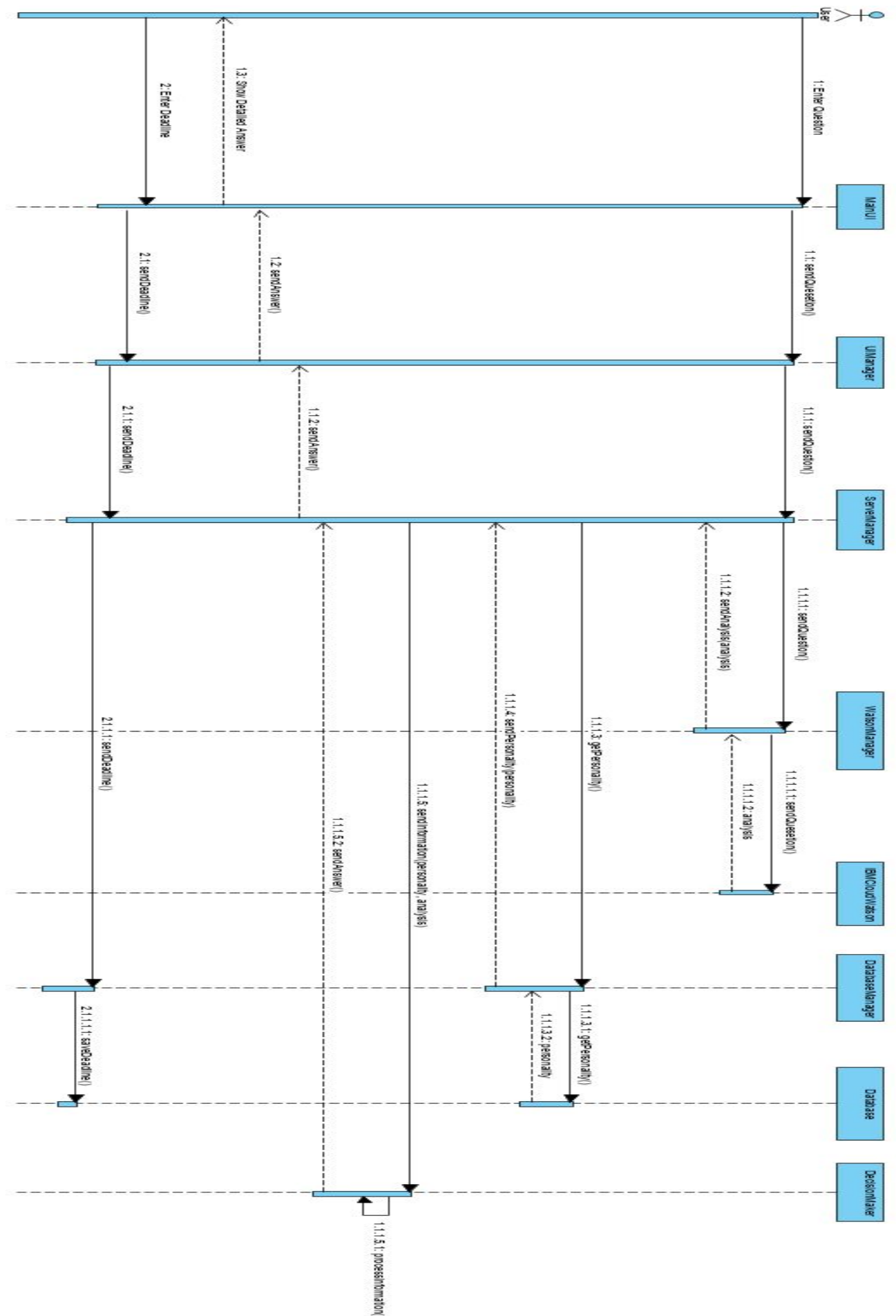


Figure 5. Sequence Diagram for Decision Making Activity

2.5.4.2 Activity Diagram

The following is the representation of our activity diagram. The diagram illustrates the activity flow of our mobile application from the user's perspective.

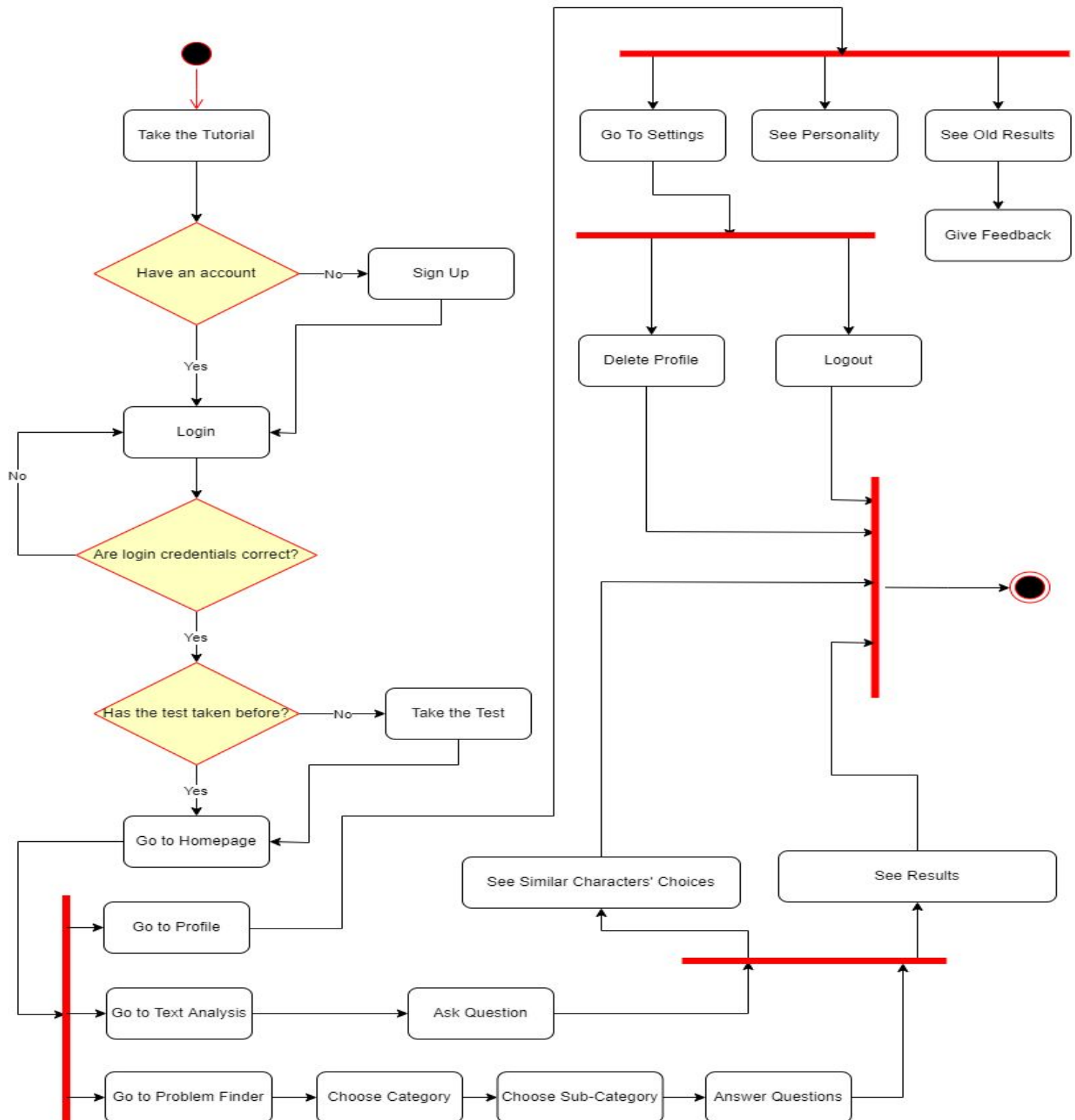


Figure 6. Activity Diagram for Csion

2.5.5 User Interface

This part contains the navigational path and the screen mockups of the application.

2.5.5.1 Navigational Path

Here, we illustrated the navigation of the application, i.e, what to click to reach any screen in application.

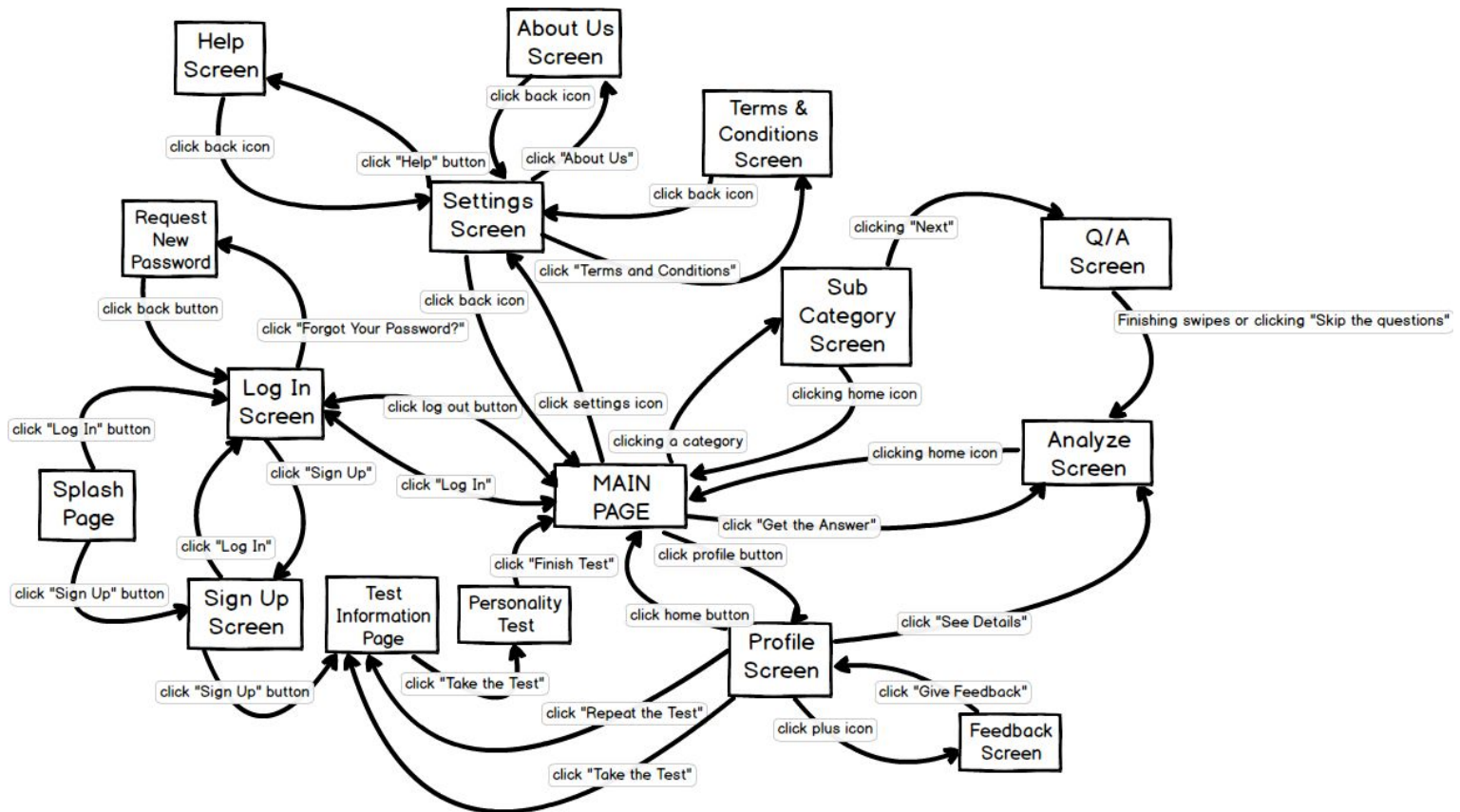


Figure 7. Navigational Path

2.5.5.2 Screen Mockups

Here, we present each screen of our application with mockup figures and explanations.

Splash Page

Splash page (see Figure 8, 1st screen) opens up when the user first downloads and runs the application. It has the logo, slogan and two buttons: If user already has an account, s/he can click *Log In* button, if not s/he should click *Sign Up* button to register.

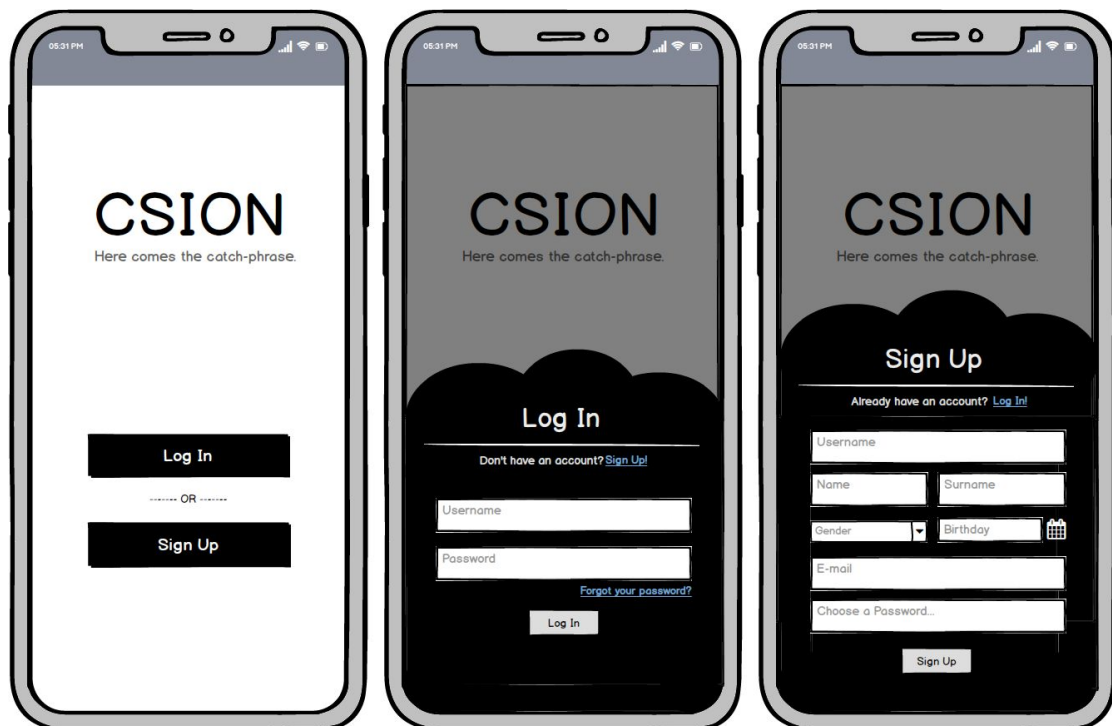


Figure 8. Splash Page

Log In Screen

After clicking the Log In button at splash page, user navigates to Log In screen (see Figure 8, 2nd screen) and by filling out the form and click Log In button, s/he can be navigated to the main page. In the case that the user navigated to this page by mistake, s/he can click *Sign Up* link to navigate to registration screen. User can click *Forgot your password?* to request new password via email through a pop-up.

Sign Up Screen

By clicking the *Sign Up* button at splash page or Log In page, user can be navigated to sign up screen (see Figure 8, 3rd screen). User should fill the form and click *Sign Up* to register our system. If user already has an account, by clicking *Log In*, s/he can be navigated to Log In screen.

Personality Test Screens

As a part of the registration, users are obligated to take a personality test. So, after clicking *Sign Up* button at Sign Up screen we bring up the test immediately starting with an informative page.

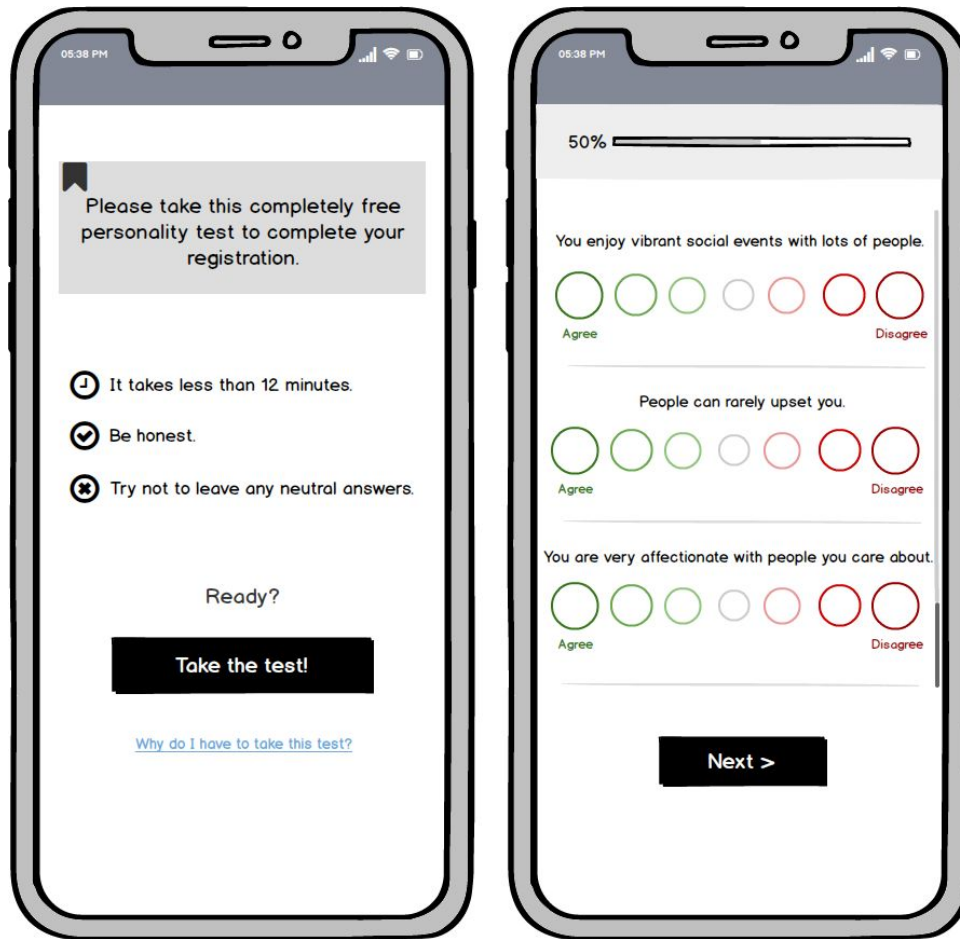


Figure 9. Personality Test Screens

Test Information Screen

At the first screen at Figure 9, we give a short information to user about test and for further detail user can click the related link to see a pop-up text having more detailed information about our motives. User should click *Take the Test* button to proceed.

Personality Test Screen

Test will have several questions that giving a rate range to user, looking like the 2nd screen at Figure 9. After each bundle of questions there will be a Next button, at the end of last question bundle, Next button will turn to Finish the Test button. After user clicks it, she will be registered and navigated to Main Page.

Main Page

This page is the core of the application, where user can navigate to all of the pages and also the core system -which is getting decision advice- lies in this page. User can get use of two different question asking systems at this page: Categorical System and Chatbot System.

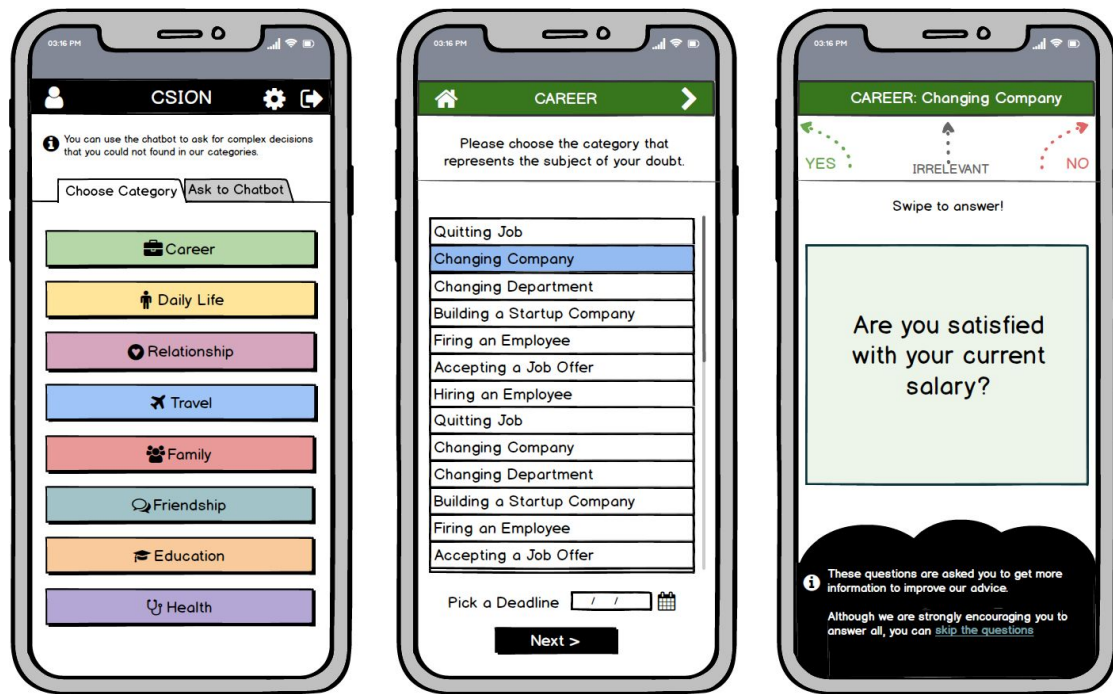


Figure 10. Main Page and Decision by Category Screens

Navbar

Starting from the main page, we will have a navigation bar at the top of the screen to keep the navigation more easy-to-use. For the purpose of explaining the buttons only once, we will mention the icons with general use at navbar here. At first, the person icon at the left top of the Main Page (see Figure 10, 1st screen) navigates to Profile page. The cogwheel icon at the right top navigates to Settings, and the icon beside it is the Sign Out button, it navigates back to splash page by ending the session. The home icon at the left top of 2nd screen at Figure 10 navigates to Main Page. The icon with an arrow showing left at navbar of some pages (see Figure 13) is a back button to navigate back to the previous screen.

Choose Category Tab of Main Page

When user clicks to this tab at Main Page the first screen at Figure 10 opens up. This part includes a list of categories and user can choose the category that is related with his/her question. This act navigates to Sub-Categories Page (see 2nd screen at Figure 10).

Sub-Categories Page

This screen (see 2nd screen at Figure 10) has a list of sub-categories for user to relate with the question in their mind. For example, if user having an indecision about quitting his/her job and having a question in his/her mind, such as “Should I quit my job?”, then s/he clicks the related subcategory, which is “Quitting Job” in that case. After that, user should pick a deadline by using the datepicker below that represents until which date s/he should decide. By clicking the *Next* button, user navigates to Q/A Screen. At navbar of this page, there is an arrow directed to right for users to be able to navigate through categories.

Q/A Screen

At this screen (see 3rd screen at Figure 10), application will ask user several questions to provide optimal analysis, however, if user does not wish to answer any questions, s/he can click Skip the questions to directly navigate to analysis screen. To answer the questions, users can swipe to shown directions. After doing the last swipe, analysis page (see Figure 11, 2nd screen) will load.

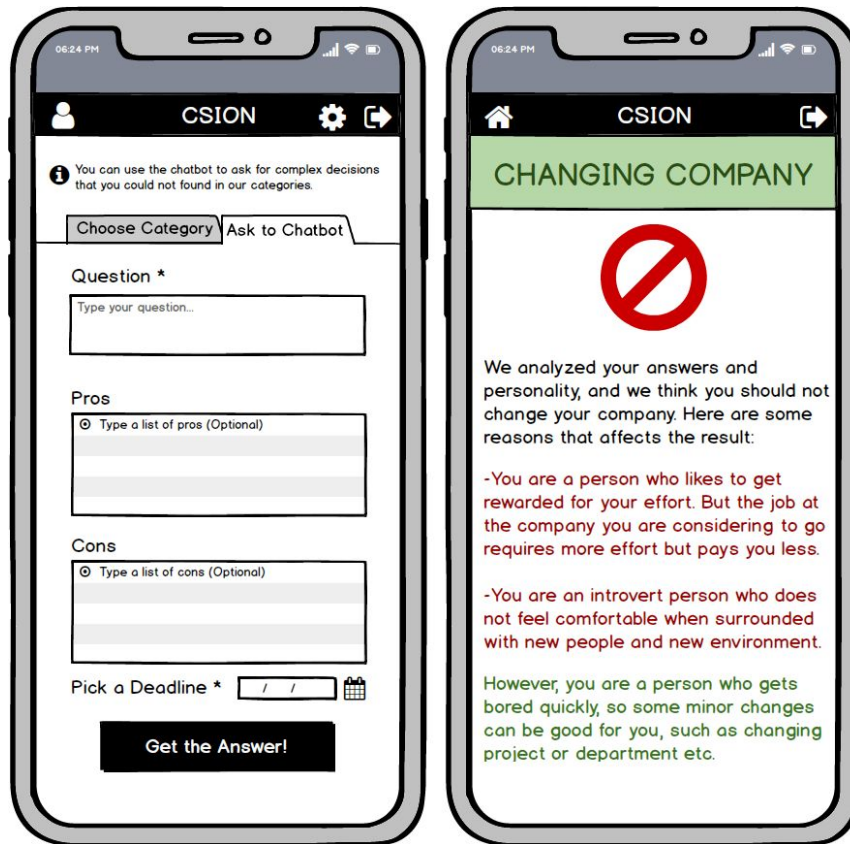


Figure 11. Main Page and Analysis Screen

Ask to Chatbot Tab of Main Page

User can type his/her more complex question that s/he could not find at categories through this tab. For this system to work, user should type an explicit question to first textbox and also should pick a deadline at datepicker. Typing pros and cons are optional. User should click Get the Answer button to navigate Analysis screen shown as 2nd screen at Figure 11.

Analysis Screen

At this screen, we offer the user our detailed analysis and give advice. There is no functionality at this page other than navbar.

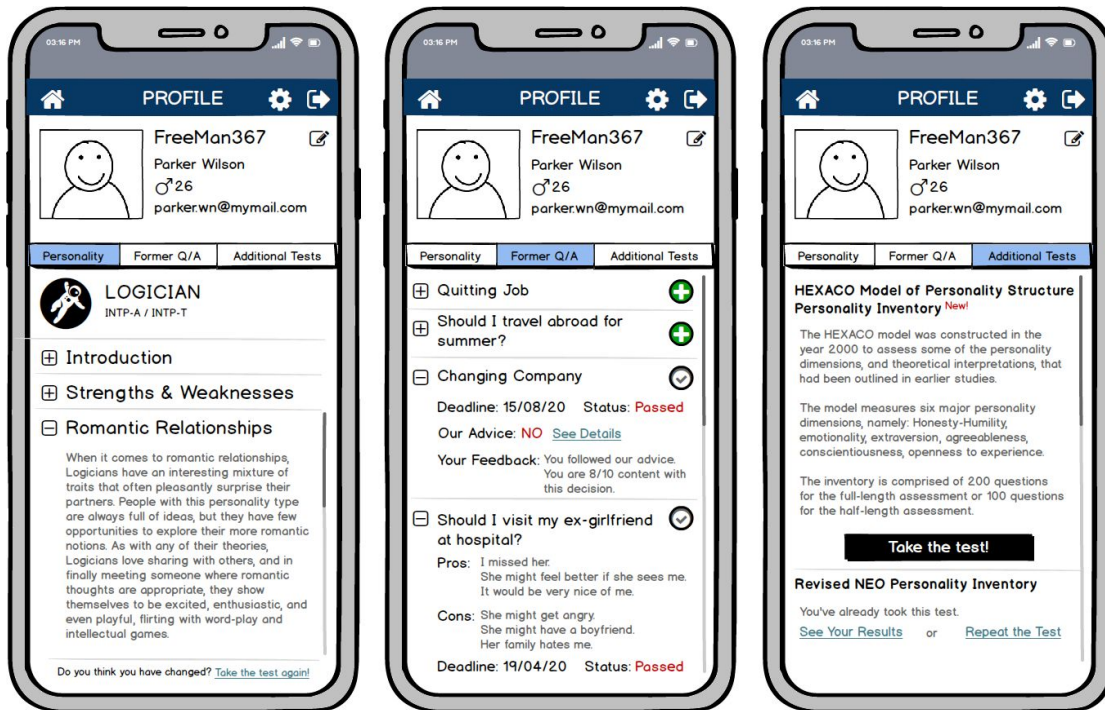


Figure 12. Profile Page

Profile Screen

At his screen, user can see and edit user information, see the personality test result by clicking *Personality* tab (see 1st screen at Figure 12), specialties of former questions s/he asked by clicking *Former Q/A* tab (see 2nd screen at Figure 12) and also give feedback to old questions s/he asked when the deadline is passed, and finally take additional tests and see their results by clicking *Additional Tests* tab (see 3rd screen at Figure 12). Clicking plus icon at *Former Q/A* tab navigates user to *Give Feedback* screen (see Figure 13), and clicking *Take the test!* or *Repeat the Test* button at *Additional Tests* tab or *Take the test again!* at *Personality* tab will navigate user to *Personality Test Screens* (see Figure 9) having the test s/he chose.

03:16 PM

← GIVE FEEDBACK

Quitting Job

i We formerly advised you to NOT to do this action.

Did you do this action?

☒ YES ☐ NO

How content are you with this decision you made?

★★★★★

You can give further details you want to share with us about your experience.

Type here.. (Optional)

Give Feedback

Figure 13. Give Feedback Screen

Give Feedback Screen

User can choose what decision they gave and rate how content they are with this decision through this page (see Figure 13). User can also give further detail about their experience by typing, however, this is an optional section. Clicking *Give Feedback* button will dave the feedback navigate user back to updated *Profile* page.

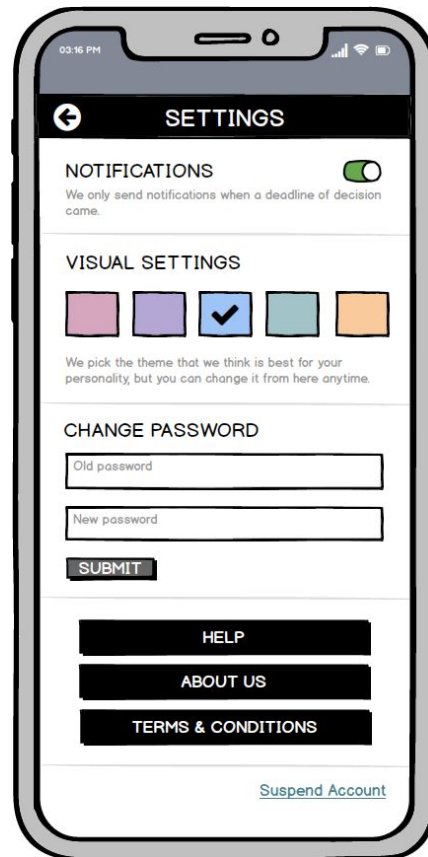


Figure 14. Settings Screen

Settings Screen

Clicking cogwheel icon at navbar will navigate user to this screen (see Figure 14). User can change the notification and theme preferences through this screen and also change password by confirming old password, typing a new password and clicking *Submit* button. User can suspend the account by clicking the related link, read further information about the application by clicking the *Help* button, read about the creators and sponsors of the application by clicking *About Us* button and finally read the Terms & Conditions again by clicking the related button.

3 Other Analysis Elements

In this part we have explored several risks and factors that affect our project. We also provide the plans we have made for the course of action we want to take in this project.

3.1 Consideration of Various Factors

In this part, we discussed and analysed how our solution can be affected by given factors, public health, public safety, public welfare, global factors, and cultural factors.

Table 1: Factors that can affect analysis and design.

	Effect level	Effect
Public health	8/10	If the algorithm is not designed properly, regardless of the filters, or any filters is forgotten or crashed, it may threaten the public health; like suicides, homicides etc. Hence, filters should be designed faultless.
Public safety	8/10	As mentioned above, if the decision mechanism crashes, it may make their users to threaten public safety incase of violating society. To exemplify, if user asks a question about spreading violence, and if the program could not understand that the question is about the one of the banned topics, it will answer it like a normal question. Hence, filters and permissions should be analyzed and developed well to avoid this kind of cases.
Public welfare	6/10	Increase in public welfare may increase the using of the application; if public welfare is well, rate of buying and downloading the application will be increased. Moreover, maybe we, the developers are able to extend program with more advanced tools. However, it does not have a significant importance in comparison with other factors.
Global factors	8/10	Unexpected changes in global factors may affect application in a bad way, since its learning and prediction algorithm may be needed to renewed.

Cultural factors	10/10	If the application is not accepted by a specific culture, this will damage its market value. Decisions that may include sensitive ideas might not be displayed. To exemplify, program should not give answers to specific topics which includes region, politics, etc
Social factors	10/10	The answering algorithm should be designed according to various social factors. Given answers should not violate any social relations. Moreover, its answering and prediction algorithm may differ in different regions' or populations' different social factors. System should be able to understand user's social relations, situations etc then give answers by taking these into consideration. If not, the program will not be preferred by the users.

3.2 Risks and Alternatives

In this part, we explained possible risks, their effect on our project and our plans to solve those issues.

Risk 1: In our project, the biggest risk is that we will try to use Natural Language Processing in our project which is a field that no group member had experience in before. Since it is an advanced topic, we did not take any classes in that field. Hence, we can not be sure if we can handle to use and customize IBM Watson to make Natural Language Processing in our application. The likelihood of this risk is moderate because, even though we do not have any experience in that field, IBM has excellent documentation for starters and there are many tutorials about how to use and customize IBM Watson on the internet. Hence, we think, we can learn the tool in the available time. If we could not handle to implement it in our project, the effect of it will be impactful because this is our project's main feature. However, just in case, we have included another suggestion mechanism that works like a decision tree that can replace our Natural Language Processing method if we can not implement it.

Risk 2: Another risk in our project is that we want to use cross-platform technologies to implement the client-side of our project because we do not want to implement two versions of it, one for iOS and one for Android. Rather than implementing two versions, we want to implement one version with cross-platform technologies and use that version in both iOS and Android. However, even though we used native technologies for mobile devices before, we have not had any experience in cross-platform technologies yet. Hence, we need to learn it from scratch before implementing our application on it which we can not be so sure that if we can handle it or not in the remaining time. If we can not handle it, it will cost us some time because we definitely start learning them and if we decide that it will not work for us, we will have to move to the native technologies to implement our mobile applications.

Risk 3: Another risk of our project is that the part where we suggest to use feedbacks that we collect from users for deep learning to improve our decision maker algorithm requires big data's to perform well. For that purpose, we are planning to start with the data coming from CrystalKnows API. However, in the case that coming data is useless for our purpose, it is very hard to collect enough feedback to feed deep learning algorithm. So, if that is the case we do not stop collecting feedbacks from user but we will use those feedbacks in a different way. Rather than learning from them, we will let the user see the statistics what other people do and if our suggestion make them happy or not. By showing this additional information, we will try to improve our application's reliability.

Table 2: Risks

	Likelihood	Effect on the project	B Plan Summary
Risk 1	Moderate	One of our main feature can not be done.	Decision tree method will be the only suggestion mechanism.
Risk 2	Low	Will cause time loss. Other than that it will not affect the project.	Changing our technology stack to native development technologies.
Risk 3	Moderate	Our algorithm can not improve itself from feedbacks.	Using feedbacks to only show additional statistics to users rather than using that data for deep learning purposes.

3.3 Project Plan

At this part of the report we explained our work packages via several tables, and draw a Gantt Chart to explain our plan for future work.

3.3.1 Work Packages

In this part, we have identified major key points to keep track of our progress throughout the project. We have divided, the total workload into smaller work packages and analyzed their requirements as sub-tasks. In each work package there is one leader who oversees the development progress of the package and decides on the course of action members should take. First two work packages cover the High and Low Level Design stages, where we create a robust architecture for the project and create detailed documentation to determine the final priorities of the project to produce High and Low Level Design reports respectively. The next package deals with the implementation of CrystalKnows API into our project and producing a primal UI for the development process during which the display quality will be improved significantly. The fourth one cover the implementation of the database and the transfer of our manager classes into a proper server to handle the control of the project. The last package deals with the usage of Natural Language Processing via a customized version of the IBM Watson machine to be able to properly interpret asked questions and create responses to the user accordingly. The end result of this stage will produce the final version of the Csion when it is connected with the rest of the work packages, and will enable us to deliver a final report of our project.

Table 3: List of work packages

WP#	Work package title	Leader	Members involved
WP1	High Level Design Stage	Ayça	All Members
WP2	Low Level Design Stage	Selim	All Members
WP3	CrystalKnows Api / UI Implementation Stage	Nursena	All Members
WP4	Database/Server Implementation Stage	Yavuz	All Members
WP5	NLP/Analysis Algorithm Implementation Stage	Mehmet	All Members

Table 4. The List of Work Packages

WP 1: High Level Design Stage			
Start date: 11.11.2019 End date: 31.12.2019			
Leader:	Ayça Begüm Taşcıoğlu	Members involved:	All Members
Objectives: To produce a software architecture for the application. Getting an insight about the general concept. Identifying the flow of control.			
Tasks: Task 1.1 <Subsystem Decomposition> : To categorizing the needs and requirements of the system Task 1.2 <Persistent Data Management> : To resolve the maintenance of the data processing Task 1.3 <Control & Security> : To specify the general control of the app and enable secure data transfer Task 1.4 <Boundaries> : To identify the limitations and scope of the project ...			
Deliverables D1.1: High Level Design Report			
WP 2: Low Level Design Stage			
Start date: 01.01.2020 End date: 23.02.2020			
Leader:	Mehmet Selim Özcan	Members involved:	All Members
Objectives: To finalize the design before the implementation stages. Creating detailed documentation. Determine the final priorities of project.			
Tasks: Task 2.1 <Trade-offs> : Identify the object design trade-offs. Setting a preferred aspects of the app Task 2.2 <Documentation> : Create the elaborate documentation for the development Task 2.3 <Final Class Designs> : Create packages and form the class interfaces			
Deliverables D2.1: Low Level Design Report			
WP 3: CrystalKnows API / UI Implementation Stage			
Start date: 01.01.2020 End date: 23.03.2020			
Leader:	Nursena Kurubaş	Members involved:	All Members
Objectives: To integrate the CrystalKnows API into our system and customize it Create a simple UI design to enable test user interactions.			
Tasks: Task 3.1 <API Integration> : To setting ready CrystalKnows API for initial user character categorization. Task 3.2 <Keyword Generation> : To identify the keywords of specified personality type Task 3.3 <Personality Test UI> : To create UI for Myers/Briggs personality test implementation Task 3.4 <User Profile UI> : To create UI for information and test results for user Task 3.5 <Homepage UI> : To provide a user-friendly UI for navigating through app Task 3.6 <Problem Detection UI> : To construct necessary appearance to ease the description of the problem Task 3.7 <Analysis UI> : To create a template for final detailed result of the algorithm			
Deliverables D3.1: Customized CrystalKnows API D3.2: Basic Csion UI			
WP 4: Database/Server Implementation Stage			
Start date: 01.01.2020 End date: 23.04.2020			
Leader:	İsmail Yavuzselim Taşçı	Members involved:	All Members
Objectives: To create a sustainable and maintainable database. To deploy the server which enables the communication amongst different subsystem of the project. To identify the database attributes.			
Tasks: Task 4.1 <Create Primal Database> : Create a database template with entities and their attributes Task 4.2 <Generate the Question Database> : Create a database for designated questions Task 4.3 <Link the Databases> : Establish the connection between the app and the primal databases Task 4.4 <Deploy the Databases> : Place the database on remote server Task 4.5 <Create Primal Server> : Create a server for handling the requests and respond accordingly Task 4.6 <Deploy the Server> : Making the server ready to operate online			

Deliverables
D4.1: <i>Csion Database</i>
D4.2: <i>Csion Server</i>

WP 5: NLP/Analysis Algorithm Implementation Stage			
Start date: 01.01.2020 End date: 15.05.2020			
Leader:	Mehmet Sanisoğlu	Members involved:	All Members
Objectives: To customize the Natural Language API for detection of the problem domain. Create keywords according to result of domain detection. To develop Analysis Algorithm to evaluate the keywords and user inputs to produce a detailed output.			
Tasks: Task 5.1 <Customize the API> : Teach the desired behaviour of the related keyword identification analysis Task 5.2 <Integrate with Server> : Placing the customized API into Server Task 5.3 <Develop an Analysis Algorithm> : Create an algorithm by considering the researches and interviews with psychology experts. Come up with an accurate scoring methodology to produce an output			
Deliverables D5.1: Customized Natural Language API D5.2: Csion v1.0 D5.3: Final Report			

3.3.2 Gantt Chart

Here, we presented our future work plan for project Csion via a Gantt Chart.

Gantt Chart of Csion

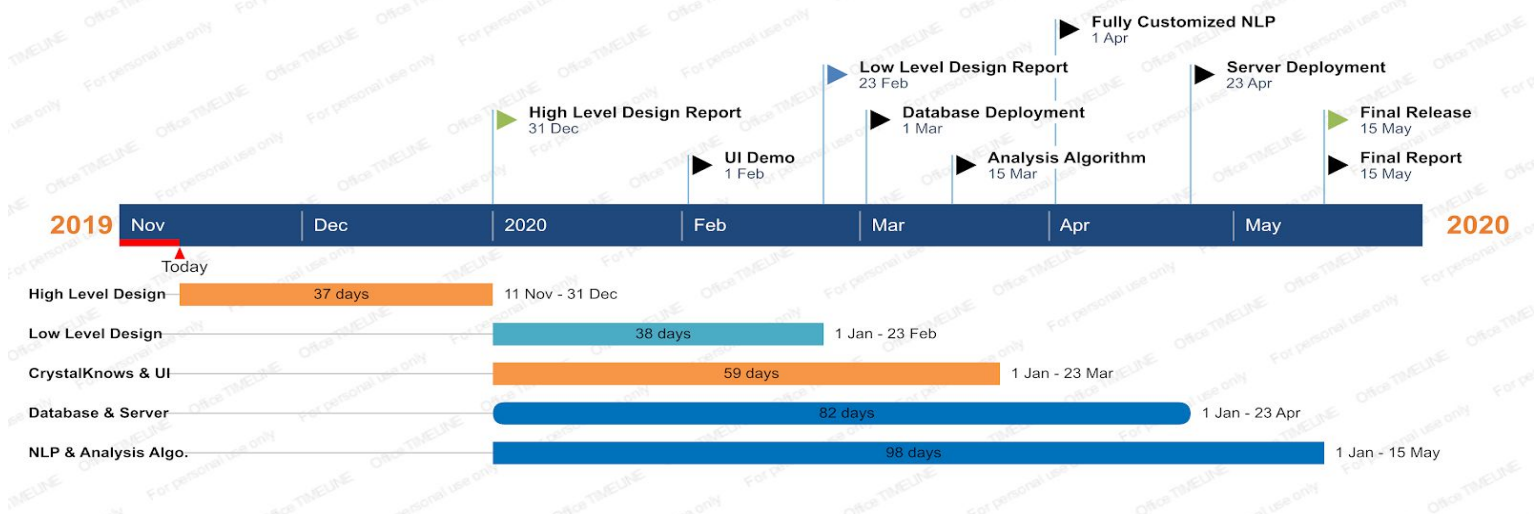


Figure 15. Work Plan of Csion

3.4 Ensuring Proper Team-Work

As a team, we have investigated many project management tools to synchronize our work as best as we can. Some of the tools we investigated was Office Timeline, Asana, Jira and Trello and we decided to use Trello, mostly because it is free. But we also recognized that Trello is more convenient to use at our meetings to track down our work, and it also seemed like visually more clean and user-friendly to us. We created our Gantt Chart by using Office Timeline to investigate it better, but since we chose Trello now, we will insert our Gantt Chart to our project at Trello.

As a repository for our source code, we decided to use GitHub, since we all have hands-on experience on how to use it and based on this experience, we thought GitHub is convenient for our project.

3.5 Ethics and Professional Responsibilities

- All collected sensitive user data from questionnaires and personality tests should be used and stored according to General Data Protection Regulation (GDPR) by following their guidelines [3].
- All collected sensitive user data should be transferred using encrypted protocols like HTTPS.
- User passwords should be stored as salted hash.
- All sensitive user data should be encrypted with a symmetric key.
- We should be transparent about data collection, data processing and we should inform our users about it.
- Consent of users should be taken before processing their data to improve our application.
- To make sure that users' data are safe, we should take database backups frequently and store those backups in different locations.
- We should respect our users' right to be forgotten and we should be able to clear their sensitive data without any problem.
- We should store and transfer third party data coming from CrystalKnows API in a safe environment to be able to protect their proprietary data.
- We should create a guideline to follow while answering user questions. Such as questions about suicidal thoughts or criminal actions or political concepts should not be answered.
- We should create a clear Terms & Conditions form and make sure that users read and accept it before using our application.

3.6 New Knowledge and Learning Strategies

Although we have not started the implementation of the project yet, we have already started to acquire new knowledge about many issues. So far, we have done many online research about the APIs that we may need, such as, CrystalKnows Personality Assessment API and many different chatbot APIs. We are investigating how to customize and manipulate chatbot APIs, how to build a cross-platform application etc. We also arranged several meetings with our innovation expert to discuss the details and feasibility of our application, and we manipulated our ideas accordingly. While making decisions about our implementation, we also share our knowledge and experience about different areas and make our design choices by considering the ideas of our peers.

For future learning, we need to learn cross-platform development better and we need to make practice on the framework we chose for this purpose. We will watch crash-course videos and make use of online resources. For the machine-learning and AI purposes that we will use the chatbot system of our application, we decided to use IBM Watson technology and we are considering to have a hands-on experience to learn it since it

seems to be more convenient for its use. At this part, we also need to understand and learn more about how natural language processing work.

We are also considering to use NodeJS for server implementation, so we need to learn and practice it by doing exercises. For database implementation, we are considering to use MongoDB or MariaDB, we will make online research, interview with experts and gather information about both of them, and then we will decide which one is more convenient for our project.

4 References

- [1] "The purpose of the Myers-Briggs Type Indicator® (MBTI®) ," *The Myers & Briggs Foundation - MBTI® Basics*. [Online]. Available: <https://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/home.htm?bhc p=1>. [Accessed: 13-Oct-2019].
- [2] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.
- [3] "General Data Protection Regulation (GDPR) Compliance Guidelines." *GDPR.eu*. [Online]. Available: <https://gdpr.eu/>. [Accessed: 13-Oct-2019].