YÜKI - SOCIAL SKILLS CHALLENGER
Capstone Report

April 2020

K. Kadaoui
Supervised by Pr. O. Iraqi
Capstone Report

**Student Statement:**

I, Karima Kadaoui, assert that I have applied ethics to the design process and in the selection of the final proposed design. And that I have held the safety of the public to be paramount and has addressed this in the presented design wherever may be applicable.

___________________________________________________
Karima Kadaoui

Approved by the Supervisor

___________________________________________________
Pr. Omar Iraqi
ACKNOWLEDGEMENTS

I’m at a loss of words to express my gratitude to all the people that have been uplifting me during these 5 years. This project would not have seen light if not for the tremendous help provided by my supervisor Pr. Omar Iraqi, the incredible work of many of my professors, Dr. Hind Saddiki for introducing computer science to me, my friends for challenging the project idea to help me identify the weak points and make it stronger (especially mocking its previous name) and most importantly my parents who sacrificed more than is possible to list.

The support I have received goes even beyond the scope of this project. I have learned so much, changed in countless aspects and discovered more about the person I want to be. It has not always been easy; I have found myself thinking too low of my capabilities or too high of myself. But the most important thing I have learned is that it always gets better. It might take days or years. You won’t realize it immediately. And one day you’ll remember how it was something you struggled with.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>5</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>6</td>
</tr>
<tr>
<td>2 FEASIBILITY STUDY</td>
<td>7</td>
</tr>
<tr>
<td>2.1 TEMPORAL FEASIBILITY</td>
<td>7</td>
</tr>
<tr>
<td>2.1.1 RESEARCH</td>
<td>7</td>
</tr>
<tr>
<td>2.1.2 IMPLEMENTATION</td>
<td>7</td>
</tr>
<tr>
<td>2.2 RESOURCES FEASIBILITY</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1 RESEARCH</td>
<td>7</td>
</tr>
<tr>
<td>2.2.2 IMPLEMENTATION</td>
<td>8</td>
</tr>
<tr>
<td>2.3 FINANCIAL FEASIBILITY</td>
<td>8</td>
</tr>
<tr>
<td>3 STEEPLE ANALYSIS</td>
<td>9</td>
</tr>
<tr>
<td>3.1 SOCIAL IMPACT</td>
<td>9</td>
</tr>
<tr>
<td>3.2 TECHNOLOGICAL IMPACT</td>
<td>9</td>
</tr>
<tr>
<td>3.3 ECONOMIC IMPACT</td>
<td>9</td>
</tr>
<tr>
<td>3.4 ECOLOGICAL IMPACT</td>
<td>9</td>
</tr>
<tr>
<td>3.5 POLITICAL IMPACT</td>
<td>9</td>
</tr>
<tr>
<td>3.6 LEGAL IMPACT</td>
<td>9</td>
</tr>
<tr>
<td>3.7 ETHICAL IMPACT</td>
<td>9</td>
</tr>
<tr>
<td>4 LITERATURE REVIEW</td>
<td>10</td>
</tr>
<tr>
<td>5 FIELD WORK</td>
<td>11</td>
</tr>
<tr>
<td>5.1 CAMPUS SURVEY</td>
<td>11</td>
</tr>
<tr>
<td>5.2 COUNSELOR OPINION</td>
<td>13</td>
</tr>
<tr>
<td>6 REQUIREMENT SPECIFICATION</td>
<td>13</td>
</tr>
<tr>
<td>6.1 FUNCTIONAL REQUIREMENTS</td>
<td>13</td>
</tr>
<tr>
<td>6.2 USE CASE DIAGRAM</td>
<td>14</td>
</tr>
<tr>
<td>6.3 NON-FUNCTIONAL REQUIREMENTS</td>
<td>15</td>
</tr>
<tr>
<td>6.3.1 PERFORMANCE</td>
<td>15</td>
</tr>
<tr>
<td>6.3.2 PORTABILITY</td>
<td>15</td>
</tr>
<tr>
<td>6.3.3 SCALABILITY</td>
<td>15</td>
</tr>
<tr>
<td>6.3.4 SECURITY</td>
<td>15</td>
</tr>
<tr>
<td>7 REQUIREMENT VALIDATION</td>
<td>16</td>
</tr>
<tr>
<td>7.1 FUNCTIONAL REQUIREMENTS</td>
<td>16</td>
</tr>
</tbody>
</table>
ABSTRACT

Social anxiety is known to be a non-negligible hindrance to one’s well-being. Since there is little to no escape from social interactions, socially anxious people live a daily battle with their fears.

This capstone project is about the app “Yūki” that aims to help people that suffer from social anxiety using exposure therapy.

Exposure therapy is known to help treat social anxiety: by exposing them to their cause of distress, patients suffering from various anxiety disorders, phobias, PTSD and OCD show improvement with their conditions.

The app will offer users real-life challenges that they need to accomplish (e.g. asking a stranger for the time). When undertaking a challenge, the user will be shown “testimonials” of other users who have previously successfully completed the channel, explaining how it went for them and giving tips if they have any. This community-based approach will simulate a support group, and the completion of the challenges will be reward-based (e.g. point system).

Keywords: Social anxiety, exposure therapy, challenge, gamification.
1 INTRODUCTION

Social Anxiety is not foreign to me. To many of my friends neither. Wherever you might be on its spectrum, it must be hindering your life in some way, due to your fears and inhibitions being the ones in control.

Through a combination of exposure therapy and a game-like environment, the Yūki app aims to repeatedly immerse users in the social situations they dread the most. Only this time, with no significant repercussions. By doing this repeatedly, users will develop the competitiveness they experience in playing games. And from that, their confidence will hopefully grow as a result.

This report will present social anxiety and what research says about it, along with how the app is planning to help. It will go over the feasibility of this project on different aspects, in addition to a STEEPLE analysis and lay down requirements for the first version of the app. It also includes different design diagrams and the different tools used in its development. Last but not least, it’ll present implementation details and features and lay down a plan for future work.
2 FEASIBILITY STUDY

2.1 TEMPORAL FEASIBILITY

2.1.1 RESEARCH
Research is needed on the following aspects:

- Social anxiety and how it differs from shyness and introversion, and whether the app could help in extreme cases where a challenge won’t help.
- Exposure therapy and how it tackles anxiety disorders specifically along with its limitations.
- How the Yūki app can simulate exposure therapy and how it can offer a personalized experience (rather than a one-size-fits-all solution).

As there are only a few things to do research on, my estimation would be that it will take about a month. Once the information gathered and the basis of the plan laid out, it will still be possible to make adjustments while in the implementation phase.

2.1.2 IMPLEMENTATION
Besides the research, time is required at the implementation level. It should be sufficient since I’m already familiar with some of the technologies thanks to my experience in class projects and my internship. Additionally, since I can refer to my supervisor while implementing the project, any confusions can be cleared out of the way in a timely manner.

Conclusion: The project is feasible in terms of time.

2.2 RESOURCES FEASIBILITY

2.2.1 RESEARCH
The plan is to use available research papers through the university’s database, along with surveying students and interviewing counselors available on campus.
2.2.2 IMPLEMENTATION

The plan is to use my personal computer, open source frameworks and free online tutorials and documentation.

The following are the software tools that are to be used for the completion of this project.

- Front End: Vue.
- Back End: NodeJs & ExpressJs.
- Database Management: Firebase.
- Hosting: Firebase.
- Version Control: Git.
- WebStorm will be used as the IDE.

Conclusion: The project is feasible in terms of resources.

2.3 FINANCIAL FEASIBILITY

All of the resources mentioned in both the Research and Implementation sections are available free of charge, and my WebStorm license expires on May 13th at which point I should be done with the implementation.

Conclusion: The project is feasible in terms of finances.
3 STEEPLE ANALYSIS

3.1 SOCIAL IMPACT
The social impact is a major one for the Yūki app: the level and access to education, social anxiety levels of the people and their attitudes towards e-commerce and self-development are all social factors that will have a direct impact. Social-media influence and traditional media are also non-negligible determinants.

3.2 TECHNOLOGICAL IMPACT
The technological impact is also extremely significant: advances in technology mean opening a window of opportunities for development and overall enhancement of the app. A better internet infrastructure around the world also means more accessibility to the app.

3.3 ECONOMIC IMPACT
Since there is a paid plan, the state of the economy and the purchasing power of consumers will affect the business, along with inflation rates and levels of spending.

3.4 ECOLOGICAL IMPACT
There seems to be no significant ecological impact on the app.

3.5 POLITICAL IMPACT
There is little political impact relating to freedoms (e.g. right to internet access).

3.6 LEGAL IMPACT
Since there is a paid plan, overall tax and fiscal policy can impact the app.

3.7 ETHICAL IMPACT
There’s an ethical impact on the Yūki app in the form of Transparency and Data Protection regulations.
LITERATURE REVIEW

Social anxiety is a trait that is characterized with nervousness and stress in social situations [1]. It is a fundamental aspect of life and when moderate, “fear of embarrassment may serve to increase arousal and attention to social interactions, inhibit aggressive or inappropriate social behavior, and help motivate preparation for social performances or presentations” [1].

Social anxiety becomes problematic when it’s a source of either distress or impairment. An example of distress would be when you obsess about an event such as a presentation or important meeting and impairment is when you turn down opportunities or plans to avoid social interactions [2].

Although social anxiety as a trait is different from Social Anxiety Disorder, as the latter is pathologic and has a threshold criterion of functional impairment, “The signs and symptoms of social anxiety disorder are similar to those of normal state or trait social anxiety and shyness.” [1]. Shyness also brings confusion as to where it stands but it is important to note that it is a lay term, making its definition problematic [1]. However, according to Schenier, “support for shyness being closely related to Social Anxiety Disorder—but without the disorder’s diagnostic threshold—comes from reports of a high degree of overlap (94%) between shyness and diagnosis of social anxiety disorder in treatment-seeking individuals (who would be expected to be the most severely shy)” [1]. What this means is that a lot of people who self-identify as “shy” but do not meet the threshold of having a social anxiety disorder, are still inhibited enough to seek help and treatment. This category of people is the target audience of the Yūki app.

Since social anxiety has similar symptoms to Social Anxiety Disorder, we can safely assume that treatments for the latter (that do not involve medication) can help people with the former as well. The first line treatment for the disorder according to Pilling is Cognitive Behavioural Therapy [3], which refers to “the class of interventions that are based on the basic premise that emotional disorders are maintained by cognitive factors, and that psychological treatment leads to changes in these factors through cognitive (cognitive restructuring) and behavioral (e.g., exposure, behavioral experiments, relaxation training, social skills training)
techniques” [4]. For the time being, the Yūki app will be focusing on the behavioral aspect, and try to simulate Exposure Therapy.

During exposure therapy, patients are faced with experiences and learn to manage fear-related stimuli [5]. They are incited not to avoid the stimuli, to participate in social situations and test debilitated expectations, falsifying them in the process [6].

5 FIELD WORK

5.1 CAMPUS SURVEY

I have conducted an online survey at the level of the AUI campus. In it, I have asked people (1) whether they think they have social anxiety, (2) to rate certain tasks as easy, tricky or hard and finally (3) ask if there are other situations that weren’t mentioned that they find difficult. 154 responses were recorded.

What I was hoping to achieve through the survey was:

- Assess the need for the app in the first place from the number of people whose lives are affected by social anxiety.

- Identify which situations are most problematic to people and which are less, and that to start with the ones who are easy to overcome and get into more challenging experiences as the users gradually build their confidence.

- Get more data about what people are struggling with and make challenges from it.

The results of the first question turned out to be interesting: the majority of people couldn’t evaluate whether they suffer from social anxiety, and the ones who didn’t think so only made up 17.5% of the responses.
Although the sample was quite small, and might not accurately reflect the real campus situation, the 127 people who answered “Yes” or “I might be” seem to me to be enough of a motive to make the app.

The second question however, varied immensely in terms of results. Even the situation that gathered the most “Easy” ratings (113), namely “Asking for ketchup or tissues at the counter” still had 41 “Tricky” or “Hard” ones (28 and 13 respectively). It is thus clear that no social situation seems to be unanimously easy (or hard), and there is also no striking pattern as to what the challenge order to follow should be. Generating a random order is also not an option as it can potentially discourage some users that might struggle greatly with the first challenges and quit before they build the resilience they need to develop the habit.

Even though the last question only got answers from about a third of the subjects (46), many of them gave multiple scenarios, thus making the total number of situations much higher than 46.
5.2 COUNSELOR OPINION

After getting the survey results, I had a meeting with Aure Veyssière to ask her about her opinion on the project since she is a counselor and clinical psychologist with a Ph.D. in clinical psychology & psychopathology.

She liked the idea of the app but warned me about the use of the “social anxiety” term, as it is usually confused with other terms such as shyness and introversion. She also mentioned that the condition should be handled very carefully and that pushing someone beyond their capacity can worsen the situation, and it’s easy to do that when you’re not an expert in the field. That being said, we went over the challenges that I had in mind and she approved of them.

After I told her about my concerns about the challenges’ order, she confirmed that it is indeed important and should be taken into consideration. I thus suggested that upon signing-up, a user would be prompted to answer a sort of quiz or test where they would rate situations similar to how it was done in my survey, and from those answers a tailored and unique path would be created to ensure the effectiveness of the app.

Veyssière finally suggested adding the counselors’ office number somewhere in the app to be available whenever things might get particularly tough for a user, which I thought was a brilliant idea.

6 REQUIREMENT SPECIFICATION

6.1 FUNCTIONAL REQUIREMENTS

- A user shall be able to register using their email, Facebook or their Google account.
- A user shall be able to login using their email, Facebook or their Google account.
- A user shall be able to log out of the app.
- A user shall be able to fill a test where they rate their level of comfort with different social situations.
- A user shall be given a tailored “tree” of challenges based on their test responses.
- A user shall be able to “undertake” a challenge, i.e. make it the current active one.
- A user shall be able to drop a challenge from the active state.
- A user shall be prompted to write a testimonial on their experience after a successful completion of a challenge.
- A user shall be able to view other users’ testimonials on a challenge they’re undertaking.
- A user’s score shall be increased after a successful completion of a challenge.

6.2 USE CASE DIAGRAM

![Use Case Diagram](image)

Figure 6.2.1
6.3 NON-FUNCTIONAL REQUIREMENTS

6.3.1 PERFORMANCE
- The app should not exceed 2s while booting up.
- The app should not exceed 1s to return results and render.

6.3.2 PORTABILITY
- The app should run on all hardware and operating systems.

6.3.3 SCALABILITY
- The performance should not be affected with an increasing user base.

6.3.4 SECURITY
- Confidentiality
  - The system should protect sensitive data and prohibit unauthorized access.

- Integrity
  - Data should be authentic, accurate and free of corruption.

- Availability
  - The app should be available at least 98% of the time per month.
## 7 REQUIREMENT VALIDATION

### 7.1 FUNCTIONAL REQUIREMENTS

| Requirement                                                                 | State                      |
|                                                                            |                           |
| A user shall be able to register using their email, Facebook or their Google account. | Using email: Met  
Using FB/Google: Not met |
| A user shall be able to login using their email, Facebook or their Google account. | Using email: Met  
Using FB/Google: Not met |
| A user shall be able to log out of the app.                                | Met                        |
| A user shall be able to fill a test where they rate their level of comfort with different social situations. | Met                        |
| A user shall be given a tailored “tree” of challenges based on their test responses. | Met                        |
| A user shall be able to “undertake” a challenge, i.e. make it the current active one. | Met                        |
| A user shall be able to drop a challenge from the active state.            | Met                        |
| A user shall be prompted to write a testimonial on their experience after a successful completion of a challenge. | Met                        |
A user shall be able to view other users’ testimonials on a challenge they’re undertaking.

A user’s score shall be increased after a successful completion of a challenge.

### 7.2 NON-FUNCTIONAL REQUIREMENTS

<table>
<thead>
<tr>
<th>Requirement</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>The app should not exceed 2s while booting up.</td>
<td>Met</td>
</tr>
<tr>
<td>The app should not exceed 1s to return results and render.</td>
<td>Unmet: Still unsure on what takes long during some loads</td>
</tr>
<tr>
<td>The app should run on all hardware and operating systems.</td>
<td>Met</td>
</tr>
<tr>
<td>The performance should not be affected with an increasing user base.</td>
<td>Unsue</td>
</tr>
<tr>
<td>The system should protect sensitive data and prohibit unauthorized access.</td>
<td>Met</td>
</tr>
<tr>
<td>Data should be authentic, accurate and free of corruption.</td>
<td>Met</td>
</tr>
<tr>
<td>The app should be available at least 98% of the time per month.</td>
<td>Unsue</td>
</tr>
</tbody>
</table>
8 SYSTEM ARCHITECTURE

Figure 8.1
9 DESIGN

9.2 SEQUENCE DIAGRAM
9.3 CLASS DIAGRAM

Figure 9.2.1

10 TECHNOLOGICAL ENABLERS

**VueJS**: VueJs is an open-source front-end Javascript framework. I’m using it along with the Vuex and the Vue-router libraries (a state management pattern serving as a centralized store for components [7] and the official router of the framework respectively). I chose Vue as a front-end framework after getting familiar with it during my internship.
**NodeJs & ExpressJs:** NodeJs is an open-source run-time environment for Javascript. It is built on Chrome's V8 JavaScript engine and allows the execution of Javascript outside of the browser [8]. I chose it because it would allow me to use Javascript in both the client and server sides.

ExpressJs is an open-source framework for Node used for building web applications and APIs. [9]

**Progressive Web App (PWA):** A Progressive Web App is an app that is delivered through the web but that offers capabilities similar to those of a native application, such as being available offline, being installable and having a splash screen. Its perks are that it can be found in a search engine, that it doesn’t require a store and that it doesn’t occupy space the way a native app does.

To be able to do that, a PWA uses a Service Worker which is a Javascript component that sits as a proxy between the browser and the network. During the first load of the app, the service worker caches the resources in the browser’s cache and checks it on future loads before checking the network, allowing an offline experience.

![Diagram of Service Worker](image)

**Figure 10.1**

A PWA also uses what is known as a Manifest, which is a config JSON file where you specify things like a home-screen icon, the name to be displayed and screen orientation. [10]
**Firebase:**
- Cloud Functions: a serverless framework that lets you automatically run backend code in response to events triggered by Firebase features and HTTPS requests.
- Cloud Firestore: a NoSQL document cloud database. Firestore uses real-time listeners and provides offline support
- Firebase Authentication: supports multiple types of authentication (email, social network accounts)

### 11 IMPLEMENTATION DETAILS

#### 11.1 DISCOMFORT TEST

The discomfort test is what allows the app to classify the challenges for a user. The test gives situations that the user rates as either “Easy”, “Tricky” or “Hard”. Each situation is related to a “type” that is assigned that rating (1, 2 or 3) as a difficulty.

For example, a user might rate “Asking a stranger for directions” as “Easy” and “Asking a friend for a favor” as “Hard”. In that case, the type “strangers” will be assigned a difficulty of 1. Similarly, the “bothering” type will be assigned a difficulty of 3.

Upon completion of the test, a “discomfort map” is computed mapping between the types and their respective difficulties or discomfort levels and then stored in the database.

#### 11.2 CHALLENGE PATH

Each challenge in the app has a difficulty level and one or more types. For example, a challenge that asks a user to ask a stranger for a favor will have both the “strangers” and “bothering” types. After the discomfort map is obtained, we generate the “challenge path” for that user.

To do that, we first compute the “total difficulty” of each existing challenge. This total difficulty is calculated by adding the difficulty level of that challenge and the maximum discomfort level of its types. The individual difficulty level of a challenge helps sorting challenges that have the same types but where one is harder than the other. It doesn’t take into
account the user’s discomfort with its nature and is decided on by comparing the challenges against each other.

The discomfort is calculated by finding the max level among the types of the challenge. E.g. a challenge that has the “strangers” and “bothering” types will have a maximum discomfort of 3 for a user that rated at least one of the two types as “Hard”.

The challenge path contains rounds and each round contains challenges that have the same total difficulty. The user has the choice to start with whichever challenge they want in that round. Using rounds is beneficial since being assigned only one challenge before moving to another one might discourage users, especially when they’re just starting. It’s helpful when they have the feeling that they can start with an “easier” one.

```javascript
// Initialize variable for the smallest difficulty we should start with and highest one we should stop at
let minDifficulty = 0;
let maxDifficulty = 0;

let rounds = [];

querySnapshot.forEach(doc => {
  const challenge = doc.data();
  let maxDiscomfort = 0;
  challenge.types.forEach((type) => {
    maxDiscomfort = Math.max(map.types[type], maxDiscomfort);
  });

  const totalDifficulty = challenge.difficulty + maxDiscomfort;

  // A difficulty can never be 0, if found it means it's not set yet
  minDifficulty = minDifficulty === 0 ? totalDifficulty : Math.min(minDifficulty, totalDifficulty);
  maxDifficulty = Math.max(maxDifficulty, totalDifficulty);

  if (totalDifficulty in rounds) {
    rounds[totalDifficulty].challenges.push(challenge.id);
  } else {
    rounds[totalDifficulty] = {
      challenges: [challenge.id],
      completed: 0
    };
  }
});
```

**Figure 11.2.1**
The minimum and maximum difficulties are also added to the challenge path, the former
determining the current round that the user will start with and the latter keeping track of the last
round (the one with the maximum difficulty) since the challenge path object is not sorted.

```javascript
try {
    await admin.firestore().collection('challengePaths').doc(map.uid).set(
        {
            uid: map.uid,
            currentDifficulty: minDifficulty,
            maxDifficulty,
            rounds
        });
}

} catch (error) {
    console.log("DiscomfortController: Failed to store challenge path in db:" + error.message);
    return res.status(502).send({
        error: "Something wrong happened in our servers"
    });
}
```

**Figure 11.2.2**

11.3 TAKING AND DROPPING A CHALLENGE

Taking and dropping a challenge is simply done by setting the user’s activeChallenge
attribute to the challenge ID or setting it to an empty string.

```javascript
async take(req, res) {
    const {id, uid} = req.body;
    try {
        // Setting this challenge as the active one for our user
        await admin.firestore().collection('users').doc(uid).update({
            activeChallenge: id
        });

        return res.sendStatus(200);
    } catch (error) {
        console.log("ChallengeController: Failed to update activeChallenge:", error.message);
        return res.status(502).send({
            error: "Something wrong happened with our servers."
        });
    }
}
```

24
11.3 COMPLETING A CHALLENGE

When the user completes a challenge, they’re prompted to enter a testimonial. After submitting it, the testimonial info is stored in the database, the activeChallenge attribute set to an empty string, and the completed challenge added to the user’s array of completed challenges. (Figure 11.3.1)

After that, we increase the completed attribute for that round and check whether there are still challenges left in it. If so, our job is done. Otherwise, we check whether there are rounds left by comparing the current difficulty and max difficulty. If the two are equal, we set the current difficulty to -1, meaning that the user has completed all of the available challenges. If not, we keep increasing the current difficulty until we find an existing round. (Figure 11.3.2)
```javascript
// Add the new testimonial to db
admin.firestore()
  .collection('testimonials')
  .doc(testimonialId)
  .set(
    {
      uid: testimonialId,
      challenge,
      user: username,
      content: testimonial,
      date: admin.firestore.Timestamp.now(),
      score: 0,
      comments: []
    })
  .then(() => { // Update activeChallenge and completeChallenges for that user
    admin.firestore()
      .collection('users')
      .doc(uid)
      .update(
        data: {
          activeChallenge: '"
        ,
        completedChallenges: admin.firestore.FieldValue.arrayUnion(challenge)
      })
  })

// Get this user's challengePath
admin.firestore()
  .collection('challengePaths')
  .doc(uid)
  .get()
  .then(doc => {
    let challengePath = doc.data();
    let { currentDifficulty, maxDifficulty } = challengePath;
    let completed = challengePath.path[currentDifficulty].completed + 1;
    const challenges = challengePath.path[currentDifficulty].challenges;

    // If we finished the challenges under this round, find the next difficulty
    if (completed === challenges.length) {
      // Case where we're already at the last round
      if (currentDifficulty === maxDifficulty) {
        challengePath.currentDifficulty = -1; // TODO Handle at the level of the view
      } else {
        // Case where we should go to the next round
        for (let i = currentDifficulty + 1; i <= maxDifficulty; i++) {
          if (i in doc.data().path) {
            challengePath.currentDifficulty = i;
            break;
          }
        }
      }
      completed = completed;
    }
  })
```

Figure 11.3.1

Figure 11.3.2
11.4 FETCHING CHALLENGES

To get the challenges of the ongoing round, we check the current difficulty. If it’s equal to -1, it means that there are no challenges left and we should display a message instead.

```javascript
const challengesIDs = pathDoc.data().path[currentDifficulty].challenges;
let challenges = [];
let processed = 0;

// Get completed challenges under our user to only display the uncompleted ones
admin.firestore().collection('users').doc(uid).get().then((userDoc) => {
  const completedChallenges = userDoc.data().completedChallenges;

  // Iterate over the ids and push the respective challenge in the challenges array
  completedChallenges.forEach((id) => {
    admin.firestore().collection('challenges').doc(id).get().then((challengeDoc) => {
      // Throw error if no challenges are found
      if (!challengeDoc.exists) {
        console.log('ChallengeController: Challenge not found although challengePath exists');
        return res.status(404).send({
          error: 'No challenge was found. '
        });
      }

      if (!completedChallenges.includes(challengeDoc.data().id)) {
        challenges.push(challengeDoc.data());
      }
    });
  });
  processed++; // Async function, we need to ensure all items were processed.

  if (processed === challengesIDs.length) {
    return res.status(200).send(challenges);
  };
})

}).catch((error) => {
  console.log('ChallengeController: Failed to fetch challenges from db: ', error.message);
  return res.status(502).send({
    error: 'Something wrong happened with our servers.'
  });
});
```

Figure 11.4.1
11.5 FETCHING TESTIMONIALS

```javascript
// Getting all testimonials for this challenge
admin.firestore().collection('testimonials').where('challenge', '==', id).get().then(testimonialSnapshot => {
  testimonialSnapshot.forEach(callback: testimonialDoc => {
    let testimonial = testimonialDoc.data();
    let date = testimonial.date.toDate();

    let strDate = date.getHours().toString().padStart(2, '0') + ':' +
    date.getMinutes().toString().padStart(2, '0') + ' ' + date.getDay() + '/
    date.getMonth() + '/' + date.getFullYear();

    testimonial.date = strDate;

    testimonial.comments.forEach(comment => {
      date = comment.date.toDate();

      strDate = date.getHours().toString().padStart(2, '0') + ':' +
      date.getMinutes().toString().padStart(2, '0') + ' ' + date.getDay() + '/
      date.getMonth() + '/' + date.getFullYear();

      comment.date = strDate;
    });

    data.testimonials.push(testimonial);
  });

  return res.status(200).send(data);
}).catch(error => {
  console.log('Failed to fetch testimonials:', error.message);
  return res.status(502).send({
    error: 'Something wrong happened with our servers.'
  });
});
```

Figure 11.5.1
12 IMPLEMENTED FEATURES

The homepage of the Yūki app contains a Login and a Register button that link to their respective pages as shown below:

Figure 12.1

Figure 12.2
Upon registration, the user is given the discomfort test.

![Register Form](image1)

![Discomfort Test](image2)

**Figure 12.3**  
**Figure 12.4**

After submitting the test, a challenge path is computed for the user and the challenges under the first round are shown.
After clicking on Take, the testimonials under that challenge, and the comments under them are displayed.
After clicking on Complete, a modal prompts the user to write a testimonial.

Figure 12.5 shows the user profile with their personal info and score.

The More Help page (lifeguard ring icon) shows resources that the user can refer to when they need more help.
13 FUTURE WORK

13.1 SHORT-TERM

When I started developing this app, my expectations of what I would be able to achieve during this time period were higher than how it actually turned out. Below are some of the things that I need to work on as soon as the semester ends:

- Add a sorting feature to challenges, testimonials and comments (by date etc.)
- Add the ability to upload files to testimonials and comments (images mainly).
- Add spinners when the data is loading and have good interfaces for errors (rather than a simple red label).
- Display different More Help Numbers depending on the localization of the user.
- Create an admin/manager side to start adding challenges & modifying the test through it.

13.2 LONG-TERM

- Add levels.
- Make the More help section more than a single page (by clicking on “view more” on a category”.
- Use affiliate links in the resources that aren’t free (e.g. books).
- Add “Premium” challenges, such as events similar to speed-dating (2 strangers talking for 2 minutes, then switching up partners etc.) but that are actually “drills”, or “simulations” where pairs act out situations that scare them (e.g. one person can play a bank teller and the other one a client). The particularity of Premium challenges is that the app provides you with a situation to complete them rather than you having to go out and try to find an occasion that matches with your challenge requirements.
- Potentially create native app versions.
- Learn about game theory and what makes users keep playing games despite their challenging nature.
- Potentially use (anonymous) data for social anxiety research from giving assessments to users at certain intervals.
- Add the cognitive aspect of Cognitive Behavioral Therapy with the help of experts in the field (through video tutorials for example)
14 CONCLUSION

At the end of this journey, I ended up meeting most of the requirements although the list wasn’t very ambitious in the first place. I intend however to continue working on it and keep adding features and tweaking details until it’s something that I could potentially pitch to investors or a company. I’m not planning for that to be anytime soon as I’ve noticed while I was working on this app that I still have a long way to go to have a good understanding of all of the elements involved in the development of apps. I feel however, that the experience has been even more beneficial than an internship. Since I’m genuinely interested in the idea and trying to make it the best it can possibly be, I tend to make more research than when I’m working for someone else.

I also gained a little more discipline than the usual since I don’t have a clear idea of how much time each task could take me. I could still do better though, and that is why I’m determined to carry on the work after the semester ends.
15 REFERENCES


