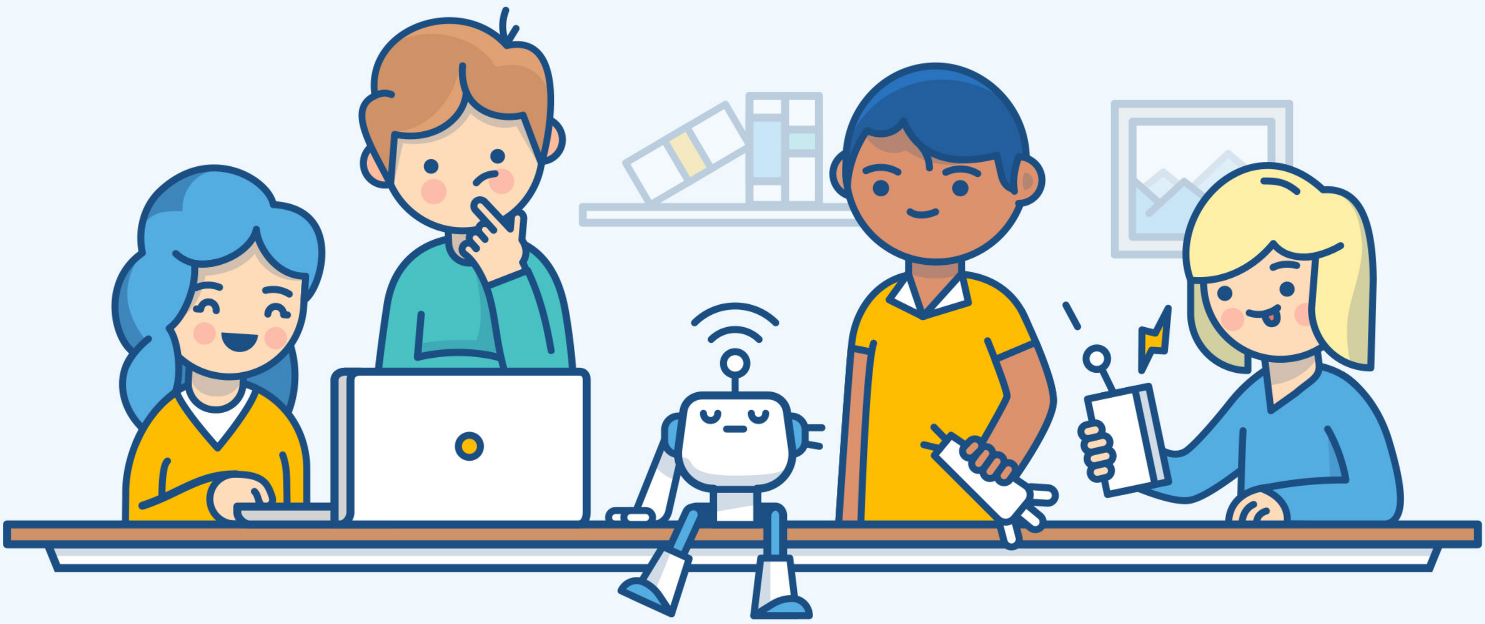
Project Plan

Stereotypes in Computer Science

Group 10D



Project Plan

Stereotypes in Computer Science

by

Alexandru Manolache (4834518),
Alin Dondera (4934245),
Andrei Geadau (4850076),
Dragos Vecerdea (4794117),
Ionut Constantinescu (4808916)

TU Delft

Contents

1
Introduction

1.1. Context

The gender gap in computer science is a major cause of concern in today's society. Nowadays, due to various factors, the field has become to be comprised of mostly males. On one hand, this means that women lose out on one of the most lucrative job opportunities. On the other hand, the field of computer science has a lot to lose with its lack of a diverse environment. A report by the National Center for Women & Information Technology [1] showed that having more gender diverse-teams in technological departments results in an improvement in employee performance, and leads to teams being able to deliver on time and under budget.

It therefore seems that it is in everybody's best interest to address the factors that contribute to this gender gap in order to possibly mitigate them. With that said, a recent study [2] claims that the earliest forecast for gender parity in computer science will be around the year 2100. Considering the importance of the situation, it is therefore important to dedicate resources in order to analyze and solve the problem.

According to [3], one of the factors that contribute to the low gender diversity in computer science is represented by the stereotypes that revolve around programmers. Studies have shown that girls who hold strong stereotypes about programming have lower interest and self-efficacy in programming which in turn leads to them choosing a different career orientation[4]. Stereotypes begin at an early age [5] and it's therefore imperative that enough attention is devoted to finding out if children hold this kind of stereotype. However, so far studies have been inconclusive. While one study shows that primary school children don't hold stereotypes about computer scientists [6], another study [7] shows the opposite. In our project, we aim to fill this gap by researching if children hold stereotypes about programmers and whether they can be influenced by a virtual intervention with role models.

1.2. Project goals

Our project consists of 2 main parts. The first part is about **data collection**. We want to develop an application that allows the client to see if children hold stereotypes against programmers. The other is about **data dissemination**. For this, we want to make people aware of their own stereotypes.

1.2.1. Data Collection

For the data collection, we aim to meet the following 3 goals.

1. **Research stereotypes that children hold about computer scientists**
   In order to meet this goal we will develop an application that collects data from children aged 6-18 years, in order to see if they hold any stereotypes towards computer science. The application will be used on laptops at the Nemo Science Museum. The children will have to complete an Implicit Association Test (IAT) to test their implicit beliefs. They will also answer different questions to test their explicit beliefs. Before the test, the parents must complete a consent form, in order for the data to be collected.

2. **Research whether stereotypes are affected by a virtual intervention with role models**
   After completing the test, the application will show children a video intervention to see if their stereotypes about computer science can be influenced. After the intervention, the children will take the test again to see if any change occurred in their beliefs.

3. **Research whether visual and textual instruments yield similar results**
   The tests will contain questions that measure both their implicit and explicit beliefs. As we previously stated, different studies have produced different results regarding children's stereotypes. While one uses textual instruments to measure children's beliefs [6] and finds no stereotypes, the other uses visual instruments [7] and concludes that they indeed hold stereotypes. By combining both types in our application, we want to find if the results from the two sources produce the same results.

1.2.2. Data Dissemination

For the data dissemination we have one goal in mind: to make people aware of their stereotypes. For this part of the project, we aim to develop an easy to use web application where a person can take an IAT about gender in computer science in order to see if he or she holds any stereotypes. A quick search on Google shows that currently there are no IATs that specifically focus on this area, the closest one focusing on sciences in general, not computer science. We aim to fill this gap with the application. Possible extensions include testing for other stereotypes about computer scientists such as hobbies or working alone vs working in a group.

1.3. Our goals

This project allows us to work on a real-world problem. We aim to use this opportunity to gain a better perspective of the software development process and get ourselves acquainted with aspects we haven't considered before, such as legal and ethical aspects. Moreover, we aim to familiarize ourselves with the latest trends and technologies in terms of web applications, while also developing a better understanding of things we already know.

1.4. Structure

In this project plan, we will start by defining the problem and analyzing some of the studies already done on the matter in Chapter 2. In Chapter 3 we analyze the feasibility of our project. In Chapter 4 we define the main functionalities of our final product in terms of the MoSCoW method and Requirements Engineering. We will then mention what we believe is the best of course of action in developing the final product in Chapter 5. Lastly, we will present a rough outline of our plan for the remainder of the project in Chapter 6.

2

Problem analysis

Gender inequality (also referred to as gender bias, gender inequity or gender gap) is the effect on individuals caused by the natural (biological) differences between men and women, but accentuated by cultural influence [8].

Computer science, despite being one of the largest and continuously growing fields, currently faces a major gender gap. Although concrete numbers for the world-wide situation are difficult to estimate, the situation can easily be summarized by recent surveys within the European Union (Figure 2.1).

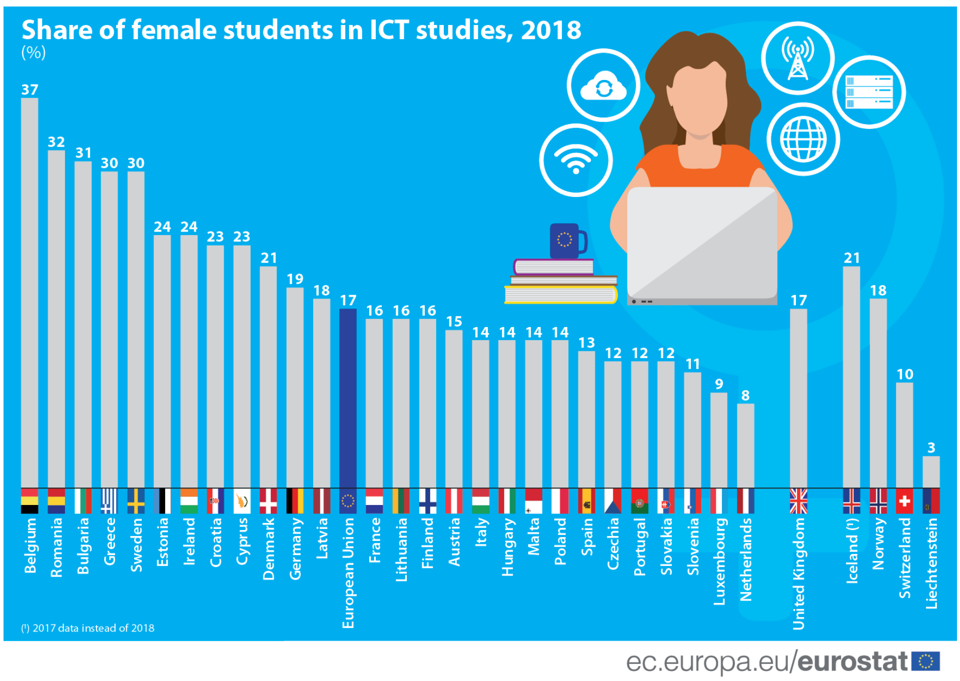


Figure 2.1: Share of female students in ICT studies 2018, from [9]

Camp et al.[10] offers a more in-depth overview of the situation of the world-wide gender gap, signaling the reverse trend that started as early as the 1980s (see Appendix A, Figure A.1).

This chapter aims to review the causes of gender gaps in the context of computer science. We will especially focus on the preconceived opinion that individuals of age 6-18 have, as well as elucidate the socio-economic factors caused by such gender discrimination. The main methods of the paper include gathering and analyzing scientific papers that already conclude various reasons on the topic [11] with the purpose of establishing and categorizing these prejudices.

In this chapter, we will first review the causes of the low gender diversity in computer science and reflect on the effects that oppressed individuals face, as well as to elucidate the socio-economic factors caused by such gender inequality. We will then elaborate on stereotyping as a primary cause while highlighting results from different studies. Lastly, we will elaborate on the unconfirmed nature of research on stereotypes among children and introduce the methodology used in our project.

Stereotypes in Computer Science

Kendall's 2011 paper [12] identifies the computer scientist's 'nerd identity' to be a key factor for lack of interest of certain groups (the paper is not limited to girls, but also other minorities).

The author states that the associations between 'computers' and nerds have a much deeper aspect, with major influences in music, cinematography, mass-media, gender, and race. To quote the article, the following five points summarize the nerd identity, which teenage girls often find repulsing:

1. Computers are an important but problematic type of technology.
2. Nerds understand and enjoy computers.
3. Those who understand and enjoy computers are nerds.
4. Nerds are socially inept and undesirable.
5. Nerds are white.

Women in Computer Science

Understanding the factors that prevent girls from enrolling in introductory computer science courses requires a deep understanding of how stereotypes affect girls, even from an early age. Recent studies emphasized the importance of the sense of belonging in female's interest in computer science [13]. Since interest is a critically important motivational variable that can affect subsequent learning performance [14], it is crucial to understand the way stereotypes shape it, in order to narrow the gender gap in the field of computer science. ' Various experiments have shown that ambient factors have a statistically relevant impact on girls' choice to enrol in STEM courses. In an experiment conducted by Master et al. [13], high-school students were shown photographs of two different classrooms, one stereotypical and one non-stereotypical, in which the same computer science-oriented course would have been carried. While girls were more likely to choose the non-stereotypical classroom, there was no preference in boys' choices. At the same time, girls were less concerned with negative stereotypes (ex. boys perform better in STEM courses) when showed the non-stereotypical environment.

This experiment, among others, paints the same consistent picture of the role of stereotypes in the girls' decreasing interest in computer science. Although there exist other variables involved in the enrollment choice, such as students' expectations for success and subjective task value [15], evidence that class environments could be a way of engaging female gender students in computer science-related courses should be a point of further discussion for change in educational institutions.

Stereotypes in children

Stereotypes held by children greatly influence their career development process. During different stages of children development, changes might occur in the way they describe occupations. In a study by Borgen and Young [16], when describing occupations, elementary school children were more likely to consider activities and behaviors while older children focused more on interests, aptitudes, and abilities. Studies, in fields like mathematics, have shown that math-gender stereotypes are acquired as early as elementary school and they "differentially influence boys' versus girls' self-identification with math prior to ages at which differences in math achievement emerge" [17].

It seems that gender might also have an influence on the type of occupation aspired by the children. It has been found that boys tend to aspire to more physically active, concrete and practical occupations, while girls have a preference towards more people-related, artistic and data-based occupations [18]. Research by Bandura et al. [19] has also determined that boys show greater confidence in aspiring to scientific and technological occupations, while girls are more confident when it comes to occupations in education, health and the social services [18].

From the ideas and studies that we have considered until this point, there seems to be a general assumption that children, regardless of age, have some kind of occupational gender stereotypes. However, for computer science-related fields, current studies have been inconclusive. In one recent study by Aivaloglou and

Hermans [6], during an eight-week Scratch programming course, children in elementary schools were asked to complete a questionnaire in order to measure their belief on four stereotypical traits of computer scientists: singularly focused, asocial, competitive and male [6]. The results of the study showed that the children did not hold any particular stereotypes and were therefore not influenced by them.

At the same time, a study by Hansen et al. [7], where children were asked to draw a computer scientist produced different results. In the study, children were asked to draw a computer scientist before and after being exposed to a computer science curriculum. The results showed that before seeing the curriculum, students viewed computer scientists as scientists who use computers, who are predominantly male and who perform their tasks alone. After the students have finished their programming course, they drew a computer scientist for a second time. Perhaps one of the most notable changes is that after the curriculum, the number of drawings of female computer scientists increased, which could be a result of the fact that the course was taught by a female teacher who could act as a role-model. This result could help us address the low gender-diversity in computer science since it shows that one of its main factors, stereotyping, can be influenced by positive experiences in the field to the extent that girls can better self-identify as computer scientists.

Identifying stereotypes in children

Implicit association tests (IATs) represent a way to measure implicit correlation and biases of a target group of people towards a certain aspect. Compared to a self-reporting study, where subjects fill out a form to answer explicit questions about a topic, in an IAT the subjects take part in a more complicated test process meant to observe their automatic associations by measuring reaction times with regard to a certain subject [20].

Harvard University provides an online platform called 'Project Implicit', where anyone can take implicit association tests. For us, it provides good examples which can be used to create our own IATs.

In one study [21], three implicit association tests conducted on female college students to examine "implicit gender-math stereotyping (greater male-math than female-math association), implicit gender identification (greater self-female than self-male association), and implicit math identification (greater self-math than other-math association)". The tests were given on a computer where the screen was split vertically in three parts. A word would appear in each part, and the participant had to assign the middle one to either one of the left or right category. By pressing '5' key, the participant assigned the middle word to the category shown on the right. By pressing 'a' key, the participant assigned the middle word to the category shown on the left. Each IAT consisted of 5 stages. The first two stages introduced two pairs of words to be shown in the left and right sides of the screen (e.g, 'them' and 'self', 'male' and 'female'). In the next stage, the two categories were superimposed, such that middle word had to be assigned to be 'them or male' or 'self or female'. Stage 4 and 5 consisted of switching the 'a' and '5' keys either for the entire test or for random parts of it. The purpose of superimposing is to record response times for the various 'combined' terms. These response times, together with the responses of the questions, can be used to draw conclusions regarding the topic of the test.

Therefore, IATs can also be designed as a way to document this type of stereotype for children. They could be modeled as an interesting and fun activity that would make them want to take such s test with ease.

Conclusion. How to solve the stereotype problem?

Current research shows stereotypes influence one's choice for a career. An undergraduate student of the female gender is less prone to consider Computer Science as a viable career path because of the ideas held about this field [22]. Therefore, we should increase research regarding Computer Science stereotypes of young students, to identify and correct the stereotypes from an early age.

In our approach, we will be studying the stereotypes of children aged between 6 to 18 years old, by building a secure, reliable progressive web-application where they can record their beliefs about computer scientists. More technical details will be discussed in Chapters 4 and 5.

3

Feasibility Study

The following chapter contains a detailed analysis to determine weather our project can be completed given the available resources. The following factors will be considered:

Technical feasibility

The current project can be reliably implemented using the technologies mentioned in Chapter 5. ReactJS and Flask framework constitute solid choices, that are used by numerous developers. Directly quantifiable advantages are the available documentation and future maintainability. The open-source nature of the two frameworks ensures high quality, abundant support, all without the hustle of dealing with license management.

Operational feasibility

Change in environment

The final product must operate to its full capacity in an ever-changing environment and configuration; **not** to be limited to the NEMO Museum of Science in Amsterdam. Portability is guaranteed by the client-server architecture, in which any modern computer will fulfill the function.

Users

The product is designed to give an unbiased result to the end-user. However, despite the best efforts, there can be false estimates of the user's stereotypes, due to factors such as concentration, background noise, etc. Should the user want to permanently remove their entry, there will be an email available for the user to express any complaints and suggestions.

Economic feasibility

The economic aspect of the project is entirely covered by VSNU Digital Society and COMMIT/. Therefore financial matters will not constitute a restriction for our project. A table containing all the estimated costs can be found below:

Index	Item	Cost
1	Hosted server (10€ per month * 3)	30€
2	Hosted database (5€ per month * 3)	15€
	TOTAL	45€

Since the project is designed for academic/research purposes, there will be no financial gain from the project.

Schedule feasibility

All team members agreed that the workload of 420 hours (40 hours per week; 8 hours per day) is sufficient for completing the project in a technical sense.

Delays provoked by the current COVID-19 situation may affect the release date of the product. As a consequence, there will be fewer opportunities to test the product before the official release date at the museum. Activities previously planned (testing the product at a local school before the official release date) will have to be postponed or canceled. It is difficult to estimate whether this lack of feedback will have any effects on the final product.

Legal feasibility

Since the application is designed for academic/research purposes, without any financial gains, the single regulation that applies is the European General Data Protection Regulation [23]. This will be accomplished by encrypting user's personal data; storing the least amount of information possible, informing users about their rights, and allowing them to withdraw from the study at any time by completing an online questionnaire.

The research project which uses the application will require approval from the Leiden's University 'Ethics committees' in order to approve collecting data for research-oriented projects. In case a pilot will be conducted, it may have to go through the TU Delft ethics committee as well.

4
Requirements Engineering

This section contains the product's final requirements, divided into the following categories:

Functional and non-functional

1. Functional requirements - describe what services the system should provide and how the system should react to particular inputs. Also include what the system should not do.
2. Non-functional requirements - describe constraints offered by the system (timing, process, standards etc.).

MoSCoW method: prioritizing the delivery of each functional requirement into the following four categories; every priority level has its own symbol for reader's convenience:

1. ✓ Must Haves
2. △ Should Haves
3. ▲ Could Haves
4. ✗ Won't Haves

Moreover, in order to distinguish between the two parts of our project (data collection and data dissemination), we will refer to a person aged 6-18 who takes part in the study as a *participant*, and a person who uses the data dissemination application for general use as a *user*.

4.1. Functional requirements

4.1.1. Data Collection

1. ✓ The application shall require parental consent before allowing participants of age 6-18 to take part in the study
2. ✓ Parental consent shall contain the following information: first name and last name of parent/guardian, participant's name, parent/guardian signature and date
3. ✓ The parental consent information must be stored in the database
4. ✓ All the collected data will be stored securely according to GDPR regulations (more details can be found in the non-functional requirements section, as well as on the website `https://gdpr.eu`)
5. ✓ Parents of participants must be able to complete the consent form in a different room than the one where the participant will take the test
6. ✓ After the consent form has been completed, the participant can take the test on one of the available laptops

7. ✓ After the consent form has been completed, the application will inform participants on which laptop to carry the test when a laptop becomes available
8. ✓ After the test is finished, the results are stored in our database
9. ✓ After taking the test, the participant will be subjected to a virtual intervention with role models
10. ✓ After the virtual intervention, the participant will take the test again in order to see possible changes. The results are sent again to the database
11. ✓ The application will support the following quiz format:
    (a) Binary choices (yes-no) answer
    (b) Multiple choice
    (c) Keypress for interactive binary choices (key E for left I for right)
    (d) Rating scale (Likert scale)
12. ✓ The quiz can be stopped/restarted by the managing staff by pressing a key combination in the event the participant leaves the test before ending
13. ✓ The user will not be able to view or edit previous questions while performing the quiz
14. ⚠ The parental consent form can be completed on parent/guardian mobile device (in compliance with WHO's ' Coronavirus disease (COVID-19) advice for the public')
15. ⚠ A single informed consent can be completed for multiple children with the same parent/guardian
16. ⚠ Siblings can carry the test on neighbouring laptops
17. ▲ The project owner must be able to add, remove, and edit questions by using a graphical user interface
18. ▲ The project owner can see intermediate results and analysis of the data stored in the database through a dashboard
19. ✗ The participant will be able to see the result immediately after taking the test
20. ✗ The participant will be able to create an account and revisit or redo the test
21. ✗ The participant will be able to compare their results with other friends

4.1.2. Data Dissemination

1. ✓ Once the user enters the web application it will be given the possibility to take an IAT
2. ✓ The IAT will give the users the possibility to discover if they hold any gender stereotypes about programmers
3. ✓ Multiple users can access the web application and take the test simultaneously
4. ✓ Before starting the test, the user
    (a) agrees with processing of personal data
    (b) understands that the results of the quiz are not definitive assessments of automatic association
    (c) understands that the results are highly influenced by various factors such as concentration, background noise, etc.
5. ✓ After finishing the test, the user will immediately see the results, that is if they hold any stereotypes towards computer scientists
6. ✓ After finishing the test, the user will also have the option to provide their email address, in order for the application to send the results via email

7. ⚠ If the user's results indicate an inclination towards certain stereotypes, the application will suggest a follow-up action for the user. Examples include reading an article or watching a video about computer scientists

8. ⚠ A session can be created and multiple users can take part and do the test. In the end, the moderator will be able to display an aggregation of all the users' test results

9. ⚠ A session can only be created with admin rights and has a unique code associated to it. This code is used by the participants to participate in it

10. ⚠ All the data gathered during a session will be removed when it ends

11. ▲ The moderator of a session has the following controls:
    (a) can terminate a session
    (b) can choose which aggregation to be displayed to participants
    (c) can limit the number of participants

4.2. Non-functional requirements

In this section, the term 'applications' refers to both Data Collection and Dissemination parts of the application.

Environmental requirements

1. The applications shall be developed for all major platforms: Windows (7 or higher), Mac OS X (10.8 and higher) and Linux, as well as mobile: Android (8 or higher) and iOS (8 or higher)

2. The web applications shall be compatible with all major browsers: Safari, Chrome and Mozilla Firefox with the following accessibility features:
    (a) Signing the informed consent will be supported on mobile, tablet and desktop devices
    (b) The data collection web application will be supported on tablet and desktop devices
    (c) The data dissemination web application will be supported on mobile, tablet and desktop devices

Development requirements

1. The Scrum methodology (with bi-weekly sprints and daily meetings) will be applied for developing the application

2. The application complete code source will be stored on TU Delft's Gitlab platform: `https://gitlab.ewi.tudelft.nl/cse2000-software-project/2019-2020-q4/cluster-8/stereotypes-in-cs/stereotypescs`

3. The non-graphical user interface components of the applications will have at least 75% code coverage

4. All components, methods and classes are documented properly

Security Requirements

1. User's personal information is stored securely in a database

2. The applications are resilient to the 'Top 10 OWASP vulnerabilities'

Legal requirements

1. The applications are in compliance with European Union's General Data Protection Regulation standard for storing, processing and using user's data

Usability Requirements

1. The applications should be responsive, in 99.5% of the cases the user will not experience delays of more than 1 second

2. The applications are intuitive: the interface is easy to navigate, and possible errors are clear and self-explanatory

5
Project approach

We believe that the best approach to meeting our goals and fulfilling our requirements is to build a progressive web-application using cutting edge tools and frameworks. Our proposal with regards to the general technologies used in the development process is to use React for the client-side and Flask for the server-side.

5.1. Risk analysis

Such a project implies risk factors which need to be noted and considered. Any decision taken regarding requirements, technologies used, etc. have to be well thought out, in order to identify the risk implications.

First risk is related to the programming languages and frameworks we will use. We have limited experience with designing the front end of an application with React or implementing a back end with Flask. It may take a few days to fully set up the bare-bones of the project.

Additionally, current legislation and world situation can influence the progress of the project. Our project involves input of sensitive personal data, hence our implementation must adhere to the General Data Protection Regulation (GDPR) for storing, protecting, and using that data. Moreover, data collection may be difficult due to COVID-19 related restrictions (e.g. ban of gatherings, closed museums).

5.2. Client-side

JavaScript is our preferred language for the client-side. Since all team members had previous academic experience with it, it has an active community and offers a wide range of choices in terms of frameworks, JavaScript seems to be the best option for us to develop a modern, lightweight and highly responsive web application. When it comes to client requirements, JavaScript proved again to be the right choice given that most of the libraries that are going to be used in the development process have consistent and detailed documentation, a key point in our client's long term view on the product. Moreover, JavaScript's popularity guarantees that potential difficulties can easily be overcome having available for inspiration a vast collection of helping materials.

5.2.1. Framework

With a lightweight backbone, extensive documentation, and numerous extensions available (Redux and Router are some examples), React is our choice. The use of Virtual DOM assures good performance and efficient UI state management. Moreover, React will permit both OOP and functional programming, giving us the possibility to explore both paradigms.

Given the advantages it offers: complex debug tools, linear user experience and responsiveness, we will develop the product as a single page application. This approach also perfectly matches with React.

5.3. Server-side

For the back end, we choose to use Python as our choice for a programming language. Due to the academic nature of our project, Python was the first language to pop in our minds, since it seems to be the standard language for academic use. Moreover, one of the requirements from our client was that the code is easy to understand and modify, which again, prompted us towards Python. It is also one of the most popular languages at the moment, with a multitude of frameworks and resources kept up to date by an active community. Another thing that we took into consideration when making our choice was the limited time we have to develop the project. Given that our team already has experience with Python, we believe that this choice will lead to the fastest development process.

5.3.1. Framework

Python being our preferred language, Flask seemed like a suitable choice as the framework for our back end. It is the recommended framework for building small-sized projects, because of its lightweight backbone and plethora of extensions. We have the ability to plug-and-play modules like authentication, RESTful API support or database communication support as we develop our applications. Furthermore, Flask is super-efficient and low-resource intensive which enables us to host it on a lower power and, hence, lower cost server.

5.3.2. Database

As persistent storage of the collected data is required, a database system is necessary. When selecting a database system, the first step would be to make a choice between an SQL and No-SQL system.

Data collection and analysis is an important part of our application, there might be need to query and aggregate the data frequently. As structured data is easier to manage, process, and understand and also ensures ACID properties, an SQL database would be the most suitable database type for our application. For this, two widely used relational database management systems that we already have experience with are MySQL and PostgreSQL. Either of them could be used in our current project.

To perform database operations through our server-side we will use the SQLAlchemy toolkit which can be easily integrated within the Flask framework.

5.3.3. Communication architecture

For the communication between the client-side and the server-side we will use a combination of REST and Web Sockets. RESTful API will be used for most parts of the application, while Web Sockets would be necessary to ensure bi-directional real-time communication for some parts of the application.

As the response message format, both JSON and XML formats are valid choices. JSON is more widely used and should suffice for our requirements because the types of responses sent from the back end will have a simple structure.

5.4. Quality Assurance

Development

To ensure proper code quality, a set of analysis tools can be used as guidance to format the code and document it properly. For Python, we will use *pylint* on top of which we will add the Flask plugin, which provides plenty of checking with regard to coding standards, error detection, and even allows rules customization.

We will be hosting our codebase on GitLab, which offers enough tools for us to have smooth version control for our code. CI/CD pipeline will be set up and will run application tests and manage automatic deployments of the back end.

The code will be thoroughly documented in order to facilitate the development process and to improve code maintainability, while also fulfilling one of our client's requirements.

Testing

Testing is a very important part of the development process and requires careful attention.

For testing the front end we will probably use an automated testing framework that records a workflow and replicates it. Examples of such frameworks are Selenium or Cypress. As testing the front end exhaustively is simply impossible, the expected coverage would be less than 75%.

The back end will be tested for each endpoint (unit testing), but also as a whole (system testing). To test the interaction with the database and between system components (integration testing) we will use mocking. For the back end, we aim to achieve a branch coverage of over 75%, using various Flask extensions like Flask-Testing and Python's *unittest* or *pytest*.

User testing will also be done to show whether the application's design and usability are suitable for the target audience (children age 6 - 18) and if the implemented functionalities match the desired requirements and expectations. As stated above, if the situation does not allow us to perform user testing in a real context, the remaining option is to rely on researchers and other members involved in the study (such as our client).

The testing process will also be documented.

Security

The connection between the client-side and the server-side shall be secure and messages that are being sent shall be encrypted. For this, a secure HTTPS connection shall be used. An HTTPS certificate will be acquired in this case.

Accessing the data stored in the database would require administrator privileges. This consists of login credentials that will be granted to the project owner and the development team. Users do not have any privileges to access the collected data.

For querying the database, prepared statements shall be used in order to avoid SQL Injection attacks. For further security, the application should enforce input validation and data sanitization.

6 Further planning and conclusions

For the remainder of our project, we aim to start the implementation phase as soon as the project plan is accepted by the client and the TU Delft coach. We will make use of an agile development process, using Scrum, where work will be done in sprints of 2 weeks, and meetings with our client will be done weekly. From our experience, shorter sprints (1 week) lead to stressful scenarios where the team cannot consistently achieve goals because of the short time frame. We believe that the combination of two-week sprints, along with weekly client meetings lets us deliver meaningful increments while also receive productive feedback. At the end of each sprint, we aim to have a prototype of our application ready. This development process offers us the flexibility of progressing in smaller steps in certain directions without committing ourselves to a certain idea, which in the end may not be what our client is looking for. To facilitate this development process, we will keep track of our progress by using the issue board functionality provided by GitLab.

> **GitLab link:** `https://gitlab.ewi.tudelft.nl/cse2000-software-project/2019-2020-q4/cluster-8/stereotypes-in-cs/stereotypescs`

Our schedule can be better seen in 6.1. Due to our course schedule and the final presentations, we have decided to also include a smaller sprint in the last week, which we will use to put the final touches on our product if needed. This schedule is not final and changes may still occur due to external factors.

Lastly, our team will try to adhere to this project plan as much as possible, but if we find out something in our approach is flawed or we missed something, we will adapt accordingly. In case this happens, we will reflect on said changes in the final report.

Week	Start Date	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
4.1	20/04/2020	Weekly Meeting with TA		Client meeting				
4.2	27/04/2020	*Holiday*	Weekly Meeting with TA	Client meeting				DL: project plan
4.3	04/05/2020	Weekly Meeting with TA Sprint planning	*Holiday*	Client meeting				
4.4	11/05/2020	Weekly Meeting with TA		Client meeting				End of Sprint 1
4.5	18/05/2020	Miderm Meeting Sprint retrospective Sprint planning		Client meeting Sprint review	*Holiday*	*Holiday*		
4.6	25/05/2020	Weekly Meeting with TA		Client meeting				End of sprint 2
4.7	01/06/2020	*Holiday* Sprint retrospective Sprint planning		Client meeting Sprint review			Pilot (estimate day)	
4.8	08/06/2020	Weekly Meeting with TA		Client meeting				End of sprint 3
4.9	15/06/2020	Weekly Meeting with TA Sprint retrospective Sprint planning		Client meeting Sprint review			End of sprint 4	DL: all componenets
4.10	22/06/2020	Final presentation						
4.11	29/06/2020							

Figure 6.1: Predicted schedule

Appendices

A

Supporting Material

Figure - reversing trend

Academic Year	CS
1980-81	32.5
1981-82	34.8
1982-83	36.3
1983-84	37.1
1984-85	36.8
1985-86	35.7
1986-87	34.6
1987-88	32.4
1988-89	30.8
1989-90	29.9
1990-91	29.3
1991-92	28.7
1992-93	28.1
1993-94	28.4
1994-95	28.1
1995-96	27.5
1996-97	27.2
1997-98	26.7

Figure A.1: Percentage of degrees awarded to women in CS, from [10]

Bibliography

[1] National Center for Women and Information Technology. (2014). What is the Impact of Gender Diversity on Technology Business Performance? Research Summary, [Online]. Available: www.ncwit.org/businesscase. (accessed: 27.04.2020).

[2] L. L. Wang, G. Stanovsky, L. Weihs, and O. Etzioni, "Gender trends in computer science authorship", *arXiv preprint arXiv:1906.07883*, 2019.

[3] S. Cheryan, V. C. Plaut, C. Handron, and L. Hudson, "The stereotypical computer scientist: Gendered media representations as a barrier to inclusion for women", *Sex roles*, vol. 69, no. 1-2, pp. 58–71, 2013.

[4] A. Master, S. Cheryan, A. Moscatelli, and A. N. Meltzoff, "Programming experience promotes higher stem motivation among first-grade girls", *Journal of experimental child psychology*, vol. 160, pp. 92–106, 2017.

[5] D. Farland-Smith, "Developing young scientists: The importance of addressing stereotypes in early childhood education", in *Early Childhood Education*, IntechOpen, 2019.

[6] E. Aivaloglou and F. Hermans, "Early programming education and career orientation: The effects of gender, self-efficacy, motivation and stereotypes", in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 2019, pp. 679–685.

[7] A. K. Hansen, H. A. Dwyer, A. Iveland, M. Talesfore, L. Wright, D. B. Harlow, and D. Franklin, "Assessing children's understanding of the work of computer scientists: The draw-a-computer-scientist test", in *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*, 2017, pp. 279–284.

[8] H. Y. Cheung and A. W. Chan, "How culture affects female inequality across countries: An empirical study", *Journal of Studies in International Education*, vol. 11, no. 2, pp. 157–179, 2007.

[9] Eurostat. (2018). Share of female students in ict studies, 2018, [Online]. Available: https://ec.europa.eu/eurostat/documents/4187653/10321595/Share+of+female+students+in+ICT+studies.png. (accessed: 23.04.2020).

[10] T. Camp, "The incredible shrinking pipeline", *ACM SIGCSE Bulletin*, vol. 34, no. 2, pp. 129–134, 2002.

[11] T. Greening, "Gender stereotyping in a computer science course", *ACM SIGCSE Bulletin*, vol. 31, no. 1, pp. 203–207, 1999.

[12] L. Kendall, ""White and nerdy": Computers, race, and the nerd stereotype", *The Journal of Popular Culture*, vol. 44, no. 3, pp. 505–524, 2011.

[13] A. Master, S. Cheryan, and A. N. Meltzoff, "Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science.", *Journal of Educational Psychology*, vol. 108, no. 3, p. 424, 2016.

[14] S. Hidi and K. A. Renninger, "The four-phase model of interest development", *Educational psychologist*, vol. 41, no. 2, pp. 111–127, 2006.

[15] J. Meyer, J. Fleckenstein, and O. Köller, "Expectancy value interactions and academic achievement: Differential relationships with achievement measures", *Contemporary Educational Psychology*, vol. 58, pp. 58–74, 2019.

[16] W. A. Borgen and R. A. Young, "Career perceptions of children and adolescents", *Journal of Vocational Behavior*, vol. 21, no. 1, pp. 37–49, 1982.

[17] D. Cvencek, A. N. Meltzoff, and A. G. Greenwald, "Math–gender stereotypes in elementary school children", *Child development*, vol. 82, no. 3, pp. 766–779, 2011.

[18] M. Watson and M. McMahon, "Children's career development: A research review from a learning perspective", *Journal of Vocational Behavior*, vol. 67, no. 2, pp. 119–132, 2005.

[19] A. Bandura, C. Barbaranelli, G. V. Caprara, and C. Pastorelli, “Self-efficacy beliefs as shapers of children’s aspirations and career trajectories”, *Child development*, vol. 72, no. 1, pp. 187–206, 2001.

[20] A. G. Greenwald, D. E. McGhee, and J. L. Schwartz, “Measuring individual differences in implicit cognition: The implicit association test.”, *Journal of personality and social psychology*, vol. 74, no. 6, p. 1464, 1998.

[21] A. K. Kiefer and D. Sekaquaptewa, “Implicit stereotypes and women’s math performance: How implicit gender-math stereotypes influence women’s susceptibility to stereotype threat”, *Journal of Experimental Social Psychology*, vol. 43, pp. 825–832, 2007.

[22] S. Cheryan, B. J. Drury, and M. Vichayapai, “Enduring influence of stereotypical computer science role models on women’s academic aspirations”, *Psychology of Women Quarterly*, vol. 37, pp. 72–79, 2012.

[23] Council of European Union, *Council regulation (EU) no 269/2014*, `http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1416170084502&uri=CELEX:32014R0269`, 2014.