CSE2000 Software Project

Psychological music recommendations:

Project Plan

May 2022

Delft, Netherlands

***Technical Writing* Group 2A**

Lucas Fatas
Diego Viero
Kenzo Boudier
Nathaniël De Leeuw
Daniel Puente Barajas

TU Coach: Gosia Migut
TA: Bianca Cosma
Client: Sandy Manolios

Contents

1 Problem Analysis

PhD Candidate Sandy Manolios [1] is studying values and personalities and their impact on music taste and preferences, within the scope of Cognitive Science. In order to conduct a proper experiment about these topics, a platform must exist such that it extracts personalities and values from its participants and links them to the music they listen to. In this project, we have been tasked with creating such a web-based platform that helps in the research of this idea. This website will be linked to in a Prolific post made by Sandy, which will redirect the participants that are willing to partake in this experiment to our website, where the actual evaluation will take place. Prolific will act as the gateway to find participants that will take part in our experiment.
Our task is to create a web-app that acts in 2 batches :

On the first one it needs to collect participants' taste in music through their Spotify account, then link it to the participants' personalities and values through a series of questions.

On the second one, it needs to feed them the same questionnaire to retrieve the participants' personalities and values, then check the closest participant from the first batch in terms of those parameters, propose the songs from its Spotify account and have the participant from the second batch rate them based on its taste.
These results will then be available to Sandy Manolios so that she can analyze them and draw conclusions for her PhD thesis.
It is worth mentioning that even though we are developing the system for a PhD researcher, we are not actually doing the research ourselves. We create the necessary tools for her research, but we'll never have to deal with the data that our application gathers.

1.1 Context

People can usually be classified into different personalities and values. We intend to explore those classifications and check if there is any correlation with the music preferences in it.
This project has been assigned to us by Sandy Manolios, a Ph.D. student with a master in Cognitive Science; the field of study she is focused on right now is music recommender systems. More specifically, their possible improvement with the previously mentioned personalities and values. Since these two concepts can be abstract and easily confused, we will provide some explanation for them in this section:

1. **Personality-based recommendations:**

   Personality is generally defined as the group of all the characteristics and traits that make one person different from another. Such traits are, for example, imagination, anxiety, or dutifulness.
   These traits can be divided into the six Hexaco categories of personality [2], according to modern psychology. We will use this division to quantify each participant's personality :

   - Honesty-Humility: in this category people who avoid manipulating others, do not break rules, and lead a humble life have a high score.
   - Emotionality: related to fear of physical dangers, anxiety at life, need for emotional support and empathy.
   - Extraversion: people who lead high extraversion scores are usually linked to social competence, enjoyment within social gatherings and interactions, leadership within a group, enthusiasm, energy and self-esteem.
   - Agreeableness: agreeable people are forgiving, lenient in judging others, can compromise and cooperate and are mild-tempered.
   - Conscientiousness: this personality category is the one that covers time and physical organization, organized work, accuracy and perfection in tasks, and careful decision-making.
   - Openness to Experience: people with a high score in this category enjoy the beauty of art and nature, like to research different topics, are imaginative, and become interested in different ideas and people.

These 6 dimensions include all the important traits of human nature, and are used as the defining factors. They are universally accepted and recognized by psychologists all around the globe to determine human personality, and it is the metric we will use for our personality evaluation.
Note that these dimensions range from positive and negative values to the specified characteristic, meaning that someone can be either very extraverted, meaning he/she is very social and open, or not at all extraverted, meaning he/she is shyer and more introverted. So this classification gives us insight in both ways of the spectrum for each of the outlined categories.
Note as well that there are a huge number of human traits, and all this division is able to classify all these traits into one of the specified categories.
We will use this division of the human psyche as the first part of this project: classify each participant by assigning a value for each of the categories, match participants with similar personalities and suggest music from this match.

2. Value-based recommendation

Values act as a guiding principle in our life. They are strongly associated with our morals and judgments. They are deeply immersed in the decision-making process and prepare individuals to act in a certain way depending on the consequences of those actions. There are several values that condition how we interact with our environment, but for this project we are going to take a look at Schwartz's Theory of basic values [3][4]. This theory establishes 10 main values to take into account when defining a person, that are derived from the three main needs for human existence: biological needs, social interaction, and group survival:

- Universalism: motivational goal of understanding and protecting the welfare of humans and nature.
- Benevolence: protecting and enhancing the welfare of human beings whom the individual is in contact with.
- Tradition: acceptance of ideologies and beliefs that one's tradition provides.
- Restrain: restriction of one's harmful impulses
- Safety: underlying motivational goal of harmony and safety of society, oneself and other
- Power: It involves status, prestige and power over other individuals and resources.
- Achievement: succeeding by showing competence according to society's standards
- Hedonism: gratifying one's own impulses.
- Stimulation: seeking excitement and thrill in life.
- Self-direction: defining and exploring one's own way.

These values are organized in a circular way according to the compatibility of such values:

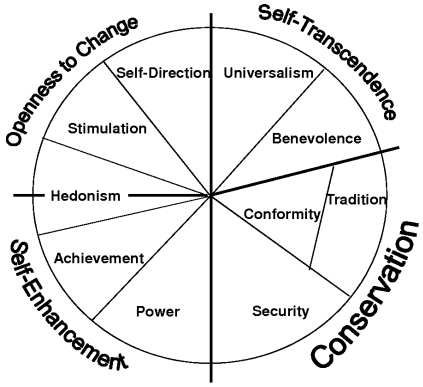

Figure 1: Schwartz's Theory of basic values

We will use the specified model for our project as well. After gathering data about participants' values, we

will match participants with similar values and recommend songs based on that.

3. **Random recommendation**

   Lastly, we will implement a random recommender in order to have a control group, which will be used as a benchmark or a point of comparison against which the other recommendations will be compared to assess the validity of the thesis.

In conclusion, we will implement these three kinds of recommendations in order to see if people with similar personalities have a more similar taste in music and check if a recommender system that uses personality and values as input for recommendation would work. We will do that by implementing the experiment in two batches: in the first batch we will just gather the personality, values and music of each participant (by making use of their Spotify account) and in the second batch the participants will as well submit their personality and get song recommendations based on that, according to the three ways described above.

We can acquire the personality and values of each participant with two simple questionnaires, and we will retrieve the participant's music taste by making use of the Spotify API and fetching the top songs of the participants. The second batch participants will then get the recommendations and provide feedback on how good such recommendations were. We also want to see if, after reviewing the songs in the second batch, they would like to add it to one of their playlist.

1.2 Stakeholders

A stakeholder is a person or organization who either influences a system's requirements or is impacted by that system. We take these stakeholders into account at the time of the requirement elicitation, so that we make sure that our project is satisfactory to the people affected by it.

There are both direct and indirect stakeholders involved in this project. The difference lies in the kind of effect that our application will have. Direct stakeholders are involved with the day-to-day activities of a project, and directly affected by the final project. Indirect stakeholders pay attention to the finished project outcome rather than the process of completing it, and are not necessarily affected by the product development.

We can then see that the direct stakeholders will be: our client, the researcher, Sandy Manolios; the Teaching Assistant assigned to our group, Bianca Cosma; the coach for the project, Gosia Migut; and us as the developers. All these people influence the daily progress of the project and are involved in the whole development process, thus they can all be classified as such.

As indirect stakeholders we can see that TU Delft will be affected by it, since one of its Ph.D. students' thesis will be influenced by this and it potentially could prove useful in the study of personalities and Cognitive Science. We also have to take into account the current privacy and internet safety laws in order not to negatively impact the participants of our website, so we have to take the participants of our webpage as passive stakeholders, since they will be influenced by the final product; and the Dutch and EU government, given that they will influence the way we handle the product.
If after the thesis publication, personality-values recommendation starts being implemented as a recommender system by big corporations, music artists should also be considered as passive stakeholders, since the number of listeners they receive on certain songs may either increase or decrease based on the recommendations.

1.3 Problem Statement

The problem given to us comprises four parts:

1. **Gather Data**
   Our group will have to conceive a website from scratch. After researching for websites, we realized that there is no similar web application available for us to build upon. This website would calculate the personality and value of a given participant based on a questionnaire. This data would be then later used for matching participants and their music taste. The challenge is that our client must be able to easily retrieve the data and manipulate it.
   The questions from which the questionnaire is composed are entirely provided by Sandy Manolios and we will display them exactly how she sent them in order not to compromise the participant analysis process.

2. **Matching Participants**
   The website should be able to match participants in three different ways; randomly which will serve as the control group, based on personality to see the correlation between personality and taste in music and lastly based on a participant's value. The matching process for personality and value will employ the nearest neighbor method to decide on a pair of individuals

3. **Recommended Songs Ratings**
   The greatest objective of our website is the retrieval of participants' opinions on a set of songs recommended through similar participants personalities and values . The client needs the data from these answers for the thesis

4. **Satisfying Stakeholders**
   During this project our main stakeholder will be our client, since a product that satisfies her needs is what we're working for. Although as we are collecting data we do need to conform ourselves to the TU Delft, Dutch and European regulation. Furthermore, we must also take into account that this project may change how recommender systems behave, impacting users and artists that use applications that implement this recommendation system.
   Our group has to find a way to encompass everyone's needs and satisfy them.

1.4 Project Goals

After all the analysis conducted in the previous sections, we have distilled our final project objective into 4 distinct sub goals specified below:

1. **Making a scalable frontend**
   We want our application to work both on computers and phones/tablets. This will allow participants to take part in the questionnaire on every device without any constraint.
   To accomplish this goal we're going to use the Tailwind[5] library, which has built-in support for scalable styling and will allow our application to be handled by multiple different devices.

2. **Using Spotify API**
   Through the use of Spotify's API we can allow participants to listen to a 30 seconds snippet of the song they're being asked about. Allowing participants to hear 30 seconds of the song they're being questioned to evaluate allows for a much better experience during the questionnaire. This will also allow for better results from which the client will be able to receive better information. The Spotify API also allow us to fetch the participants' listening history, essential in the recommendation process

3. **Easily Retrieve the Data**
   After having collected all the necessary data from the participants that answered the questionnaires, we want to create an easy-to-use platform such that the client can directly work on the collected information from there, without having to deal with stored data manually herself.

This process will facilitate the client approach to the data and the way it is handled and examined.

4. **Monitor the Experiment**
   We need to create allow the researcher monitor the experiment through the use of parameter modification (the similarity metric between users, etc) and an overview of the whole process, including control over the research by showing the size of the answers and providing a way to switch to phase 2 of the experiment with the second batch.

1.5 Inspiration from Existing Products

Our web application is partly a personality test/form and partly a music recommendation system where we ask for opinions.

For the first part, there exists a profusion of personality test websites available on the internet. There are also plenty of websites where we can create questionnaires. The most known one is Google form, but open source alternatives, like Budibase or Cryptpad, also exist. They helped us gain some inspiration regarding User Interface and how to structure the questions, showing us what ways are good to implement the front-end part of our project to make the app appealing.

These websites also helped us with ideas on how to handle the feedback after getting the recommendations.

1.6 Use cases

There are 3 different types of users that will make use of our web application: participant of batch 1, participant of batch 2 and the researcher. By Using use cases, we are pointing out their journey through our site. Based on these use cases, we can easily create functional requirements for the MoSCoW method.

Participant of batch 1 :

- The participant wants to access our site thanks to a link provided in the Prolific site.
- The participant needs to log in into their Spotify account.
- The participant needs to be able to answer questions according to personality and values, split into multiple pages.
- Our system needs to calculate the values and the personality of the participant based on the answers and store it with his/her Spotify top songs.

Participant of batch 2 :

- The participant wants to access our site thanks to a link provided in the Prolific site.
- The participant needs to log in into their Spotify account.
- The participant needs to be able to answer questions according to personality and values, split into multiple pages.
- Our system needs to calculate the values and the personality of the participant based on the answers.
- Our system matches the participant to participants of batch one with the most similar values, most similar personality and one random participant and retrieve from the database their songs.
- The participant of batch 2 listens to the songs of the matched participants of batch 1 and gives their opinion on it.

Researcher :

- The researcher wants to log in to the dashboard if she wants to change the parameters of the experiment.
- The researcher wants to retrieve the values, personality and songs of the first batch (in a CSV file).
- The researcher wants to adjust the parameters of the study (Examples: similarity metric, playlist sizes).
- The researcher wants to see the progress of the study.

- The researcher wants to retrieve the ratings of the recommended songs of the second batch participants (in a CSV file).

1.7 The Data Structure

Retrieving and processing data plays an integral role in our project which is why clearly identifying the different data structures and how they are related is crucial.

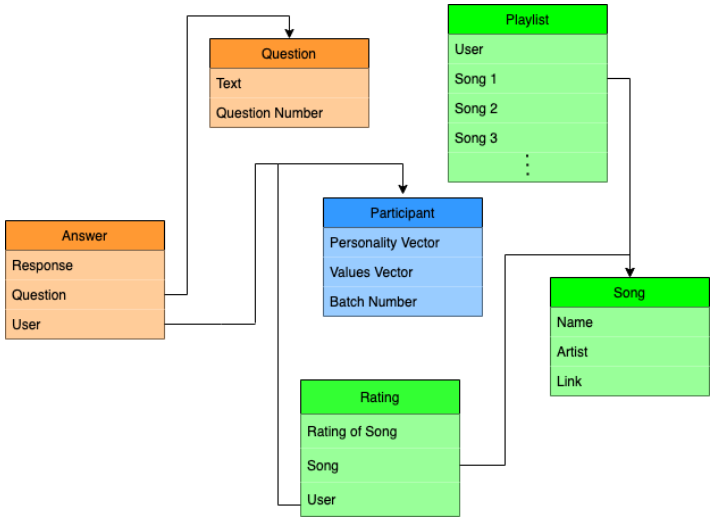

Figure 2: High level overview of data structures

1. **Participant:** The participants will have a personality vector and value vector based on their answers to the questionnaire. They are divided into two batches and will also have Spotify music data associated with it: Batch 1 participants will have its listening history attached to the vectors, and batch 2 participants will have the recommendations provided to them.
2. **Question:** Question contains a text with the question and a number to identify it. There will be three types of questions: those related to the personality part, those related to the value part. They will be provided by our client in order to effectively calculate the vectors and provide a faithful representation of both personality models.
3. **Answer:** An answer stores the response a participant gives to a question. Each answer to each question will be stored individually, and it will be structured ranging from one to five, in order to convey different levels of agreement when answering the questions -1 meaning lack of agreement and 5 meaning total agreement-.
4. **Song:** The songs retrieved from the participant's Spotify will have a name, artist and link stored.
5. **Playlist:** A collection of songs, represented in a list form. Each participant from batch 1 will have a playlist representing the listening history, and each participant from batch 2 will have three playlists attached, representing the three different kinds of recommendations provided to him.
6. **Rating:** The participant's rating of a song that was recommended to him. Each rating for each song will be stored individually.

2 Requirements Elicitation

The fact that we have only one main direct stakeholder, and not a whole company with different direct stakeholders and interests, facilitates a lot of the elicitation of the requirements. One to one interviews with our client, and discussion/brainstorming were sufficient to compose the requirements. No survey or other techniques were required.

2.1 Interview with our Principal Stakeholder

We had a first meeting with our stakeholder, namely Sandy Manolios, and her mentor Cynthia Liem before the quarter began. It was a presentation to her thesis and an introduction to the problem given to us. It helped us to have a general overview over the project and necessary context about the subject of her thesis.
Thursday 21st April 2022 we had our first official interview, where, based on the knowledge we had from the previous meeting, we were able to ask questions about uncertainties we had and also more technical parts of the project. We then constructed a first draft of the requirements.

2.2 Brainstorming Session with the Software Group

We refined the first draft of the requirements in a brainstorming session between the people of the software group, after that we had all made our own research about the subject and also existing products that are similar, for some part at least, to our product.
We were communicating with our client by Microsoft Teams posts for all remaining questions.

2.3 MoSCoW Method

We have chosen for the MoSCoW method to give prioritization to requirements [6]. While interviewing our client, we made sure to ask about the importance and priority of every requirement, to divide them easily into the categories of the MoSCoW method. While programming, we will base the sprints of the scrum methodology on these priorities.
The must-haves are the base requirements of our project. Without the must-haves the base goal of the project cannot be achieved and that is why they are going to be implemented as first.
The should-haves are in particular to improve the participant experience while being on the site, with modification possibilities and design choices.
The could-haves are mostly requirements to ease the interactions between our client and the web application. Facilitating the viewing and retrieving of data is one of these aspects.
The wont-haves are requirements that will not be implemented in the period of our project, but can possibly be implemented later on.

3 Feasibility Study

Following our initial meetings with our client, we researched the feasibility of different aspects the client wants the product to have under given conditions.

3.1 Technical Feasibility

The most important aspect we have to consider related to the technical feasibility of the product is the Spotify API. The API is integral to the product and is the technical aspect we have the least control over.
Our client would like to retrieve a participant's recommended songs on Spotify and a participant's top listened to songs over a specified time period which is possible using the API [7]. The rest of the product is straight forward regarding the technologies needed and are feasible given what we have available to us.

3.2 Schedule Feasibility

Regarding schedule feasibility, given our project requirements, we believe that completing our project within the timeframe of the course is realistic. Considering that we are a team of five working approximately 40 hours a week each, we will be able to complete our must-haves and should-haves well within the timeframe. This will allow us to be more ambitious with our could-haves.

3.3 Summary of Feasibility

After considering all the different aspects of the project, we believe that it is feasible to complete it within our timeframe.

4 Risk Analysis

4.1 Limited Real-life Working Experience

All of us, being Bachelor students, have limited real-life working experience. We've dealt with deadlines, assignments and projects throughout these 2 years, but we never worked on such an extensive and complete project from the start to the end. This may cause some delays during the process since we might not realize that a certain choice wasn't the best at a given moment. These delays could be avoided only by having a certain amount of experience on certain topics/technologies. We are aware of the implications of this project and will do our best to keep up with the (self) imposed deadlines and solve problems that may arise as swiftly and smoothly as possible.

4.2 Experience with Modern Developing Tools/Frameworks.

We decided to use libraries/frameworks for this project that are considered fairly new and were not taught during the courses that we had. Some of us already have experience with certain tools, but for the most part, working on certain sections will require a lot of documentation reading. This may slow up the development process, but it's an excellent way to learn new tools as well.

4.3 Security

Since our application will then be used in the real world by real research candidates, there's the possibility that some of them might want to fiddle with the application a bit. Because of this we must ensure that the application is safe and doesn't allow anyone to access other people's information.
The fact that, for the first batch, we are asking for Spotify username and password, even if it will be all handled by the Spotify API, adds a level of complexity security wise. We should make sure that all the confidential data goes directly to Spotify, without being changed or stored by the web application. The only personal data we should keep from the use is the API key given by Spotify, to be able to interact with it and retrieve non-sensitive information, like top songs or 30 seconds snippets of songs.

4.4 Discrimination

The diversity of the batches may be a risk point in the experiment, which may result in forms of bias towards different artists or song genres. This may negatively affect the outcome of the experiment and, if implemented within recommender systems, may be cause for discrimination and may affect the livelihoods of the people involved in the recommendations, since in the worst case some artists may get relegated and never recommended.

However, this is a risk associated with the researcher choice done by Sandy; we just need to make sure that our program works appropriately regardless of the people who take part in the experiment.

5 Requirement List

As stated above in section 2.3, we follow the MoSCoW method of requirement classification to create our requirements. We list them below in the specified fashion:

5.1 Non-Functional

- Data should be stored and handled according to the GDPR [8].
- The website must be compatible with Chrome, Firefox and Safari
- The login credentials of the researcher should be safely stored, in order to access the dashboard.

5.2 Must Have

- The participant must be able to interact with our project through a website.
- The participant must be able to express consent to share his Spotify data.
- The participant must be able to answer questions through a slider based on its preference.
- The participant must be matched with other participants based on the responses to the questionnaires.
- The participant's answers to the questionnaire must be converted to a vector representing personality and a vector representing values.
- The participant must have the personality vector, value vector and top songs stored in a persistent way.
- The participant must be able to give their opinion of songs that were recommended to him/her.
- The recommended songs will be based on the participant's listening history.
- The participant must be able to connect their Spotify account (get the information about their music preferences)
- Participant that participate in the second part of the experiment must be able to rate their recommended songs (and the batch)
- The researcher must be able to download participant data into a CSV file.
- The researcher must have a dashboard to adjust parameters for the study (Example: similarity metric)

5.3 Should Have

- The participant should be able to modify their answers
- The participant should be able to go back at any moment to the previous page
- The participant should get the music recommendations in 30 second snippets
- The researcher should be able to see the progress of the study in the dashboard

5.4 Could Have

- The researcher could be able to retrieve all the answers to the questionnaires
- The researcher should be able to like or add recommendations to his account
- The researcher could be able to see if a participant added a recommended song to his playlist
- The researcher could have an interface to sort and filter data of participants in the dashboard
- The participant should be able to access and use the website from their phone

5.5 Won't Have

- The project could be structured in a way that it will be reusable for other kinds of research in the future.

6 Project approach

The key factors we considered while deciding on the best approach for this project was our limited time frame and maintenance.

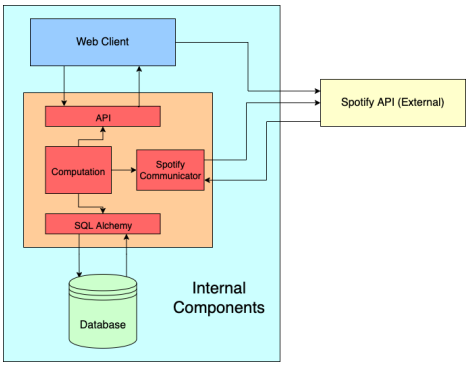

Figure 3: Diagram of Architecture

6.1 Web Client

The web client will be the front-end of our product. It will be a website, made using the React framework, available to participants and the client through the browser from a computer. The front-end will be split up into two macro sections: first batch and second batch.
The first batch will include initially a Spotify login, after which, if successful, the participant will reach the questions section. In there he/she will be able to answer questions. After the last questions the participant will submit all the questions that will be received in the backend and stored in the database with the associated participant id.

The second batch consists of participants answering the same questions as the first batch, and then, based on the given answers, specific songs will be suggested to the participant which will have to rate them based on their personal preference.

6.2 Back-End Server

The Back-End Server will be implemented in Python using Flask and it will handle the processing and computation of the data and the logic for the product. It will also handle communication with the Spotify API to retrieve the participant's songs. The Web Client will communicate with the server through a REST API that it will provide.

6.3 Database

The data will be managed using an SQL Database. The web client will not interact with the database directly, since all communication with the database will be from the Server. We will use SQLAlchemy to retrieve and update data stored in the database.

6.4 Hosting

All these technologies and resources need to be hosted somewhere. Currently, there are multiple options being considered by the client as of now. Some options involve Amazon Web Services hosting or the use of the universities hosting options. We will make a final decision with the client by week 4 to give us ample time to test the software in the hosting environment.

6.5 Testing

To ensure that our software is working as intended, extensive testing will be required for all aspects of the software. Small mistakes in our software could have detrimental effects on the study. An area of the software where this is especially true is when matching the participants. If the matching software mismatched participants, this would not be visible while using the software but would ruin the study. To ensure this does not happen, we will need a varying set of tests for the software. This set will include unit testing for the smaller components and integration testing for evaluating the compliance of the software with specified functional requirements.

7 Development Methodology

7.1 Scrum

We decided that using the Scrum with one-week sprints would be the most beneficial methodology to use for our project.

1. **Why Use Scrum?**

   While considering the different methodologies, we first decided that we would use an Agile methodology. The key reasons for this were adaptability and working software is frequently delivered. While meeting with the client, there seems to be certain aspects of the project that are still uncertain to them, so by using an Agile methodology we will have more flexibility to adapt to changes [9]. Since we have weekly meetings planned with our client, the incremental and interactive nature of agile methodologies will allow us to frequently share and get feedback on working software.

   Before settling for Scrum, another agile methodology that we considered is Kanban because of its visual and interactive methods which would enable the team to have a cohesive workflow at all times. But we decided that Scrum would be ideal for this project as we are all familiar with it , and it is better in small highly collaborative teams [10]

2. **Scrum Roles**

   For the Scrum roles, we decided that Lucas Fatas would be the project owner. He will focus mainly on managing the scrum backlog and ensuring that our software is satisfying the requirements. Diego Viero will be the Scrum master and organize our sprint meetings, manage blockers and ensure we work transparently. We will all be part of the development team.

3. **Sprints**

   We will have weekly sprints since the timeframe for our project is short. Our meetings with the client are every Thursday afternoon, so we have decided to have our sprint reviews Thursday mornings before the meetings and our sprint plannings after the meetings. This will allow us to get feedback quickly on the work we have done during the sprint and allow us to adapt to any changes requested by the client before our sprint planning sessions.

4. **Tools**

   Throughout the whole project we will make use of different tools that will help us in many areas of the process: communication, development, version control… Such tools that will be used are the following:

- Brightspace. We will retrieve all the information about the assignments and announcements from the course managers from Brightspace.

- GitLab. We will use a repository in Gitlab to share our code in a collaborative way and keep a clear overview of the version control of the code. Furthermore, we will also be using it for its usefulness in code quality checks and issue coordination. We will assign an issue for each requirement, feature and bug to be implemented or fixed in order to be able to assign a person in the group to get it fixed.

    - Our issue board will contain a list of all these issues and will also hold information about the current state of each issue in the current sprint (if the issue is to be done or is being implemented, tested, or reviewed), as well as some helpful information on how to solve said issue, which could help the developer in charge take care of it. This will give us a clear overview of our progress across sprints and a straightforward way of dividing work and checking who is doing who.
    - We will also keep another issue board with a list of the requirements, with each requirement being an issue and being assigned a MoSCoW priority. (must have, could have, should have or won't have).

    When starting a sprint, we will take a look at the issue board and choose which requirements to assign to the upcoming sprint. For every requirement, we will create sub-issues (even if there is only one) that need to be solved in order to solve the requirement, and link them to it, blocking it from closing before they are completed. These will traverse the Development board (todo → doing → testing → review).
    At the end of each sprint, we will close the requirement issues after making sure that all its sub-issues were completed. Every issue will be worked on in a separate branch and we will create merge requests when the issue is solved. We will have a protected development branch which will be our default branch. All other branches will first be merged into development, which will then be merged into the protected master branch at the end of every sprint. Merging into the two protected branches will be done by merge requests, each of which will have to be reviewed and approved by at least two other developers who have not worked on the issue.
- Campus. We will be making use of the library, Pulse and Ewi rooms to conduct meetings in person, both development meetings where we will work in the implementation of the website and meetings with our TA, client and coach. Such meetings and rooms will be booked in advance.
- Mattermost. We will be using Mattermost for communication with our TA and to keep up-to-date on the latest announcements concerning the project.
- Jitsi meet. We will use Jitsi meet to conduct meetings with our TA when it is not an option to meet on campus.
- Microsoft Teams. We will use Microsoft Teams for communication with our client and to schedule weekly meetings with her, which we will use to ask for feedback and solve some questions about the project that may arise during the development. We will use the calendar embedded in the application to organize these meetings and, when it is not possible to meet on campus, we will use Teams calls to conduct these meetings.
- Webmail: This platform will be used to maintain communication with our coach and organize the meetings with her.
- Discord. We will use Discord to send each other links and useful information to each other and handle meetings when it is not possible to meet online.
- Google Drive. We will use a shared Google Drive folder as the means to handle files about the project in an organized way. More specifically, we will tackle each assignment with a shared Google Docs so that we can all work on it at the same time. We will then export it to a PDF and submit it to Brightspace, but the Google docs document will remain in our folder for further exploration of it after submitted.

8 Reference list

[1] Sandy Manolios (n.d.). [LinkedIn page]. LinkedIn. Retrieved April 25th, 2022, from https://www.linkedin.com/in/sandy-manolios/.

[2] Lee, K., & Ashton, M. (2009). *The HEXACO personality inventory - revised*. The HEXACO Personality Inventory - Revised. Retrieved April 28, 2022, from http://hexaco.org/

[3] Schwartz, S. H. (2012). An overview of the Schwartz theory of basic values. Online readings in Psychology and Culture, 2(1), 2307-0919.

[4] Schwartz S. H., Melech G, Lehrnami A, Burgess S, Harris M, Owens V. 2001. Extending the cross-cultural validity of the theory of basic human values with a different method of measurement. Journal of Cross-Cultural Psychology 32:519-42

[5] Tailwind website with documentation : https://tailwindcss.com/.

[6] S. Hatton, "Choosing the Right Prioritisation Method," *19th Australian Conference on Software Engineering (aswec 2008)*, 2008, pp. 517-526, doi: 10.1109/ASWEC.2008.4483241.

[7] Spotify.com. (2019). *Web API Libraries | Spotify for Developers*. [online] Available at: https://developer.spotify.com/documentation/web-api/libraries/.

[8] GDPR.eu. (2019). *GDPR compliance checklist - GDPR.eu*. [online] Available at: https://gdpr.eu/checklist/.

[9] Cockburn, A., 2003. People and methodologies in software development. *Faculty of Mathematics and Natural Sciences.*

[10] Lei, H., Ganjeizadeh, F., Jayachandran, P.K. and Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, [online] 43, pp.59–67. Available at: https://www.sciencedirect.com/science/article/pii/S0736584515301599.