**MINUTES**

|  |  |  |
| --- | --- | --- |
| Date: 16.04.2019 | Time: 13:00 | Place: **PŁ, B9, room 352** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Meeting called by:** | Yurii Shcheholiev | **Note taker:** | Yurii Shcheholiev |
| **Facilitator:** | Piotr Napieralski | **Leader:** | Michał Suliborski |
| **Attendees:** | Michał Suliborski, Ania Preczyńska, Yurii Shcheoholiev, Mariusz Pisarski | | |
| **Meeting purpose:** | Receiving the tasks and roles for the project development, deciding on tools for it, discussing the steps in project design and further development process. | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Agenda item:** | Presenting the results of a research concerning interface design. Discuss steps needed for successful and convenient design. | | | | |
| Leader: | Anna Preczyńska | | | | |
| Discussion: | Ania started presenting results of her research with clarifying what actually makes an interface a good one. (From her words) the goal of user interface design is to make the user’s interaction as simple and efficient as possible. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. [Graphic design](https://en.wikipedia.org/wiki/Graphic_design) and [typography](https://en.wikipedia.org/wiki/Typography) are utilized to support its [usability](https://en.wikipedia.org/wiki/Usability).  So to create a well designed interface several steps need to be taken:   1. Functionality requirements gathering 2. [User](https://en.wikipedia.org/wiki/User_analysis) and [task analysis](https://en.wikipedia.org/wiki/Task_analysis) 3. [Information architecture](https://en.wikipedia.org/wiki/Information_architecture) (descriptive and simple information giving) 4. Prototyping 5. [Usability inspection](https://en.wikipedia.org/wiki/Usability_inspection) and testing   After getting acknowledged with those steps we started to discuss and work on an interface starting from the first point. As it was claimed before the app should support access to smartphones camera and to analyze the emotion of a person captured by it. According to such narrow functionality we concluded that the GUI itself should consists of 3-5 screens maximum.  In the next step we faced a problem due to special conditions of our users group. Because of it we cannot be sure of their behaviour and additional testing should be provided.  The third step was omitted due to low demands of our project on information optimization and because this step is directly connected to GUI prototyping.  So we moved on and spend some time on brainstorming and sketching some variants. Out of which the Ania’s one was the most professional and good looking.  In the end we concluded to work on the prototype for the next weekend using the sketches, ideas and Ania’s prototype as a base. | | | | |
| Conclusions: | The prototype of an interface should be done for the next meeting. It should be in a paper form in order to edit and modify it easily and to have a clear understanding of its general look. | | | | |
| Action items: | Creating an interface prototype | Person responsible: | Anna Preczyńska | Deadline: | 07.05.2019 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Agenda item:** | Division of roles in a team for productive work on the project. Inform the supervisor about the team decision on division of roles. State tasks for each role. | | | | |
| Leader: | Michał Suliborski | | | | |
| Discussion: | For a successful and time efficient project development it should be separated into smaller independent tasks. That is why developer teams usually separate on groups each responsible for a particular functionality. We decided to do the same, so each member is responsible for a particular part of a project basing on personal preferences and capabilities.  Ania has a good design taste and she is experienced with graphics, and photo redactors.  Michał is the most experienced programmers in a group, especially concerning Python and machine learning.  Yurii don’t have much experience in making applications, but he is good at finding special cases and potential mistakes.  Mariusz prefers to put things together and to make communication between modules and working with people. | | | | |
| Conclusions: | Ania is a designer and an artist of a team. Responsible for graphics and GUI of the project.  Mariusz is a coder and a contact person with the foundation. Responsible for user interface functions and cooperation with ASPI.  Michał is coder team leader. Responsible for face recognition and emotion computation  Yurii is a coder. Responsible for testing the application. | | | | |
| Action items: | ------------------- | Person responsible: | ------------------- | Deadline: | ------------------- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Agenda item:** | Coherently present the results on a development process steps research the topic of design patterns, software architecture development process. | | | | |
| Leader: | Yurii Shcheholiev | | | | |
| Discussion: | Yurii started with a definition of a design pattern. In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), a software design pattern is a general, [reusable](https://en.wikipedia.org/wiki/Reusability) solution to a commonly occurring problem. In other words design pattern is a best practice that a programmer can use.  For example Facade pattern: provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher-level interface that makes the subsystem easier to use.  There are alot of design patterns existing: creational, structural, behaviour, concurrency pattern families.  Some patterns only relatable to specific programming languages, which makes it difficult for us chose a specific ones. Ideally we should use as much as we can, but learn all of the patterns will require huge amount of time(there are alot of books written on this topic: Design Patterns: Elements of Reusable Object-Oriented Software, Code Complete, etc).  A good news is the most patterns are based on SOLID principles which we learned at OOP course.  Likely we have our supervisor how will help to implement and follow those patterns, but first we should think our an architecture and a clear number of functions and modules in order to be able to which pattern to use. He also recommended to get familiar with the common of them:  • Iterator • Decorator • Observer • Strategy • State • Singleton • Template • Adapter  • Façade • Flyweight • Command • Abstract Factory • Composite | | | | |
| Conclusions: | First we need to start with creating an architecture for our project and set a certain number of functions.  Get familiar with designed patterns listed above. | | | | |
| Action items: | ------------------- | Person responsible: | ------------------- | Deadline: | ------------------- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Agenda item:** | Useful tools and services for group development such as: Google Vision, TensorFlow, GitHub, etc. Review some of the most popular and efficient tools for a group of developers. Explain their favours, usage and functionality. Consult with a supervisor for his personal preferences based on work experience. | | | | |
| Leader: | Mariusz Pisarski | | | | |
| Discussion: | Development process requires a lot of afford, concentration and team communication. In order to simplify all of those tasks programming community and companies create certain APIs(application programming interface) and services to make the development process easier and faster.  Some of the most popular and useful services, libraries and API that can be used in our project:  For the efficient group work a perfect solution will be a GitHub repository.  A GitHub - a free open source repository service. It significantly simplifies work of a developers team by providing a version control, code merging, backups and quick access to the project.  Concerning a deep learning technology there are two interesting tools:Google Vision and TensorFlow library.  Google Vision - is one of the Google Cloud services which provides a powerful API and resources to work with deep learning and artificial intelligence. Because of emotion detection process need a trained machine model to work Google Vision seems to be a good solution. Unfortunately it is only partially free, but the price of usage is really small from 0.6 to 1.6 dollars.  TensorFlow - is a Python library to work with machine learning and AI. With it to build and train ML models is easy by using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging. It is also the most popular library for machine learning which means there are a lot of tutorials and forums.  So all of the discussed tools and services are a perfect choice for our project, but some additional research and learning should be done concerning Google Vision. In particular tutorial on how to use it and to train models. In addition the GitHub repository should be created. | | | | |
| Conclusions: | Create a GitHub repository, and use TensorFlow library and Google Vision in the application. Make a deeper research concerning Google Vision API. | | | | |
| Action items: | Make a deeper research concerning Google Vision API. | Person responsible: | Michał | Deadline: | 07.05.2019 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Agenda item:** | Discuss the general look of the application, functionalities and the purpose of our application. | | | | |
| Leader: | Michał Suliborski | | | | |
| Discussion: | As we see it the task of our application is to detect human face from an image and then to be able to detect the emotion expressed on the face.  In order to capable of finishing the project before the deadline we should focus on the main task - detecting emotions. From functionalities can be added, there are:   * access to different cameras of the mobile device(front and back) * have a video mode in which there is no need to take a photo * display the results of analysis (60% happiness, 23%neutral, etc).   Also because our application is a mobile one, the simpliere and more intuitive the design and the interface, the better. | | | | |
| Conclusions: | From functionalities stated the application should be capable of detecting the human face form a picture taken from front or back camera of the phone, then analyse and detect the emotion of the person. It should offer two modes: one working on a given photo, second working in realtime. Finally it should implement nice, clean and functional design. | | | | |
| Action items: | ------------------- | Person responsible: | ------------------- | Deadline: | ------------------- |

|  |
| --- |
| **Date of the next meeting:** 07.05.2019 |
| Special notes: |