**MINUTE**

|  |  |  |
| --- | --- | --- |
| Date: 07.05.2019 | Time: 13:00 | Place:**PŁ, B9, room 352** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Meeting called by:** | Yurii Shcheholiev | **Note taker:** | Yuri Shcheoholiev |
| **Facilitator:** | Piotr Napieralski | **Leader:** | Mariusz Pisarski |
| **Attendees:** | Michał Suliborski, Ania Preczyńska, Yuri Shcheoholiev, Mariusz Pisarski | | |
| **Meeting purpose:** | Discussion on the collected research materials | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Agenda item:** | Presenting the design of the application prototype | | | | |
| Leader: | Anna Preczyńska | | | | |
| Discussion: | The meeting started with Ania presenting the prototype of an interface developed. The interface was done in Adobe XD, so it was already interactive. The interface was consisting of 3 screens. First as a user launches an app, he sees a loading screen with our team logo.  After it the user sees a main menu, which has a big starting red button with yellow label on it “Start”, and a gray gear in the top right corner of the screen. If the gear button is pressed it send the user to the parameters menu. This parameters menu has checkboxes for each setting. There are 3 settings: display list of emotions detected, percentage display of analysis and live detecting. The color of the background is lightgray  The starting button from the main menu sends a user to a new screen where are the camera, the button change the camera used(front or back) and the results of the analysis made (emotion stated). This screen is dependent on the settings changed in the parameters menu.  In general the prototype matches all of the requirements, it is minimalistic, pleasant to look and uses all the stated functionalities. However, because of our target group is not a regular one, we have to check if the interface is really simple to use, also it will gives a valuable feedback about the look and maybe give some additional propositions. So, we need to contact a ASPI foundation to arrange meeting and test the interface with our target group. | | | | |
| Conclusions: | Contact with ASPI foundation and arrange meeting with the target group and shareholders. | | | | |
| Action items: | Arrange meeting | Person responsible: | Mariusz | Deadline: | 11.05.2019 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Agenda item:** | Discussing the results of team role test. Technical specification, referring to personality tests | | | | |
| Leader: | Mariusz Pisarski/ Yurii Shcheholiev | | | | |
| Discussion: | After passing a personality test at PBL classes we decided to rearrange our workflow and responsibilities in a team according to its results. From the results of the test we have 2 coordinators, 2 resource investigators, 1 shaper, 1 monitor evaluator, 2 team workers, 2 implementers and 2 specialist. To be fully developed time in all the directions, we lack planter and complete finisher. With the analysis of all positive and negative aspect of each role, it is clear that our team struggles with finishing tasks previously set and finding creative solutions for the problems occuring during the project. But in general our team roles are balanced and we complete each other. As for the technical specifications and responsibilities in the a team, there are no changes need to be done, because each member does the job where he is the best at. For example Michał as a specialist takes care of deep learning model training, Ania creates an interface, Mariusz communicates with stakeholders and Yurii coordinates and analysis the work.  Our team is motivated and focused to succeed in our application development. | | | | |
| Conclusions: | The Facecoders team is balanced and motivated to work! | | | | |
| Action items: | -------------------- | Person responsible: | -------------------- | Deadline: | -------------------- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Agenda item:** | Google Vision research, integration of it with the project. | | | | |
| Leader: | Michał Suliborski | | | | |
| Discussion: | During the research it was found out that Google Vision is integratable with Android Studio and Java language in general. In order to work with it Google Cloud platform need to be used within a created account with assigned credit card number. From the platform we need to enable an API to be able to use it in the application. The payment won’t be charged for the free trial. While creating a project we also need to add the API package to the dependencies in Android Studio and after add the <uses-permission> lines to manifest .xml file. All the instructions are to do the following are present on the team online disk.  After this procedure this procedure we will finally have an access to the APIs instructions and function.  As for the usage after looking up some tutorials and examples it is clear that to create an app detecting the object present on a camera will take just few hundred lines of code. So, the only extension we need to provide is an face selection prioritize and the analysis of it using machine learned model. | | | | |
| Conclusions: | Download Google Vision API package and use it in an application. | | | | |
| Action items: | -------------------- | Person responsible: | -------------------- | Deadline: |  |

|  |
| --- |
| **Date of the next meeting:**  11.05.2019 |

|  |  |
| --- | --- |
| **Special notes:** |  |