

Reflection Report on REVITALIZE

Team 13, REVITALIZE

Bill Nguyen

Syed Bokhari

Hasan Kibria

Youssef Dahab

Logan Brown

Mahmoud Anklis

[Reflection is an important component of getting the full benefits from a learning experience. Besides the intrinsic benefits of reflection, this document will be used to help the TAs grade how well your team responded to feedback. In addition, several CEAB (Canadian Engineering Accreditation Board) Learning Outcomes (LOs) will be assessed based on your reflections. —TPLT]

1 Changes in Response to Feedback

This section, summarizes the changes made over the course of the project in response to feedback from TAs, the instructor, teammates, other teams, and from user testers. In the final documentation changes are clearly shown by new documentation being highlighted in red (Eg. **new change**) and deletion of documentation is shown like this: ~~deletion~~.

1.1 SRS and Hazard Analysis

SRS Changes: Many components of the SRS were changed and some new components were added based on feedback. Minimal changes were made to the project constraints regarding the monetary cost and a new assumption was added. The work partitioning table was completely updated with more descriptive event names and more accurate inputs and outputs. The use case diagram was updated to include two missing pieces which are the update food and update exercises bubbles. The Functional Requirements were changed completely based on the feedback that the requirements should not state what the system should display but rather what the system should do. An Abbreviation and Acronyms section was added to the beginning of the document to help guide the user with specific naming conventions. There were minor updates to the fit criterion of the Non Functional Requirements to include the incorporation of user surveys. A new Non Functional Requirement relating to privacy and security was created.

The traceability matrices were updated with the new Functional Requirements and they were moved to the appendix. A risk was added relating to the POC demo. Finally, a new section for Anticipated and Unlikely Changes was added as well.

1.2 Design and Design Documentation

1.3 VnV Plan and Report

VnV Plan Changes: To start a lot of the system tests for the functional requirements were changed to match changes from SRS. Also from the feedback we were suggested the following "Please do not include too many implementation details but focus on the functionalities (e.g., what is your input and output) of the system. Try to think in this way: what is the input for this action button and what should the system do after I click this button? The most important thing you should care about is not what components you are using but what do you want to achieve", So we completely changed the test cases for the functional requirements to focus more on functionality, rather than implementation and also focused on the inputs and outputs to be more specific with real examples.

Also in the VnV plan it was suggested team member's roles were a bit ambiguous, so made changes in order to make it more clear what each member was testing and specifically what sections of the code / application they were testing as well. Another suggestion was to add the unit test section since MIS is now complete, in which we did with appropriate tests and traceability matrix.

2 Design Iteration (LO11)

[Explain how you arrived at your final design and implementation. How did the design evolve from the first version to the final version? —TPLT]

3 Design Decisions (LO12)

[Reflect and justify your design decisions. How did limitations, assumptions, and constraints influence your decisions? —TPLT]

4 Economic Considerations (LO23)

I feel as though there is a market for our product. It would be an easily-learned and friendly app for people looking to flip their lifestyle regarding health-related habits. I think marketing something like this could be done by using fitness-oriented internet personalities (ie. on Youtube, Instagram, etc.) to promote and speak about the product. As for the cost, I think that could be estimated to at least 20 000 dollars per month (considering all of us don't get

paid and do this for free). The software infrastructure that must be walled behind an app like this; the servers, the database, would probably cost something like 20 000 per month. I think other than monthly software-infrastructure subscription payments, there wouldn't be too much of an overhead cost to productionalize the app. Perhaps some licensing, trademarking, etc. but other than that it would be more of a time-sink on the engineers' (our) behalf, not a money-sink. We wouldn't charge for our product; that would completely render us as a non-option against all the other competitor apps which already exist on the app store as a free to use service. Instead, we would probably try to generate ad revenue. I think a good amount of users could be attracted, given that we use the right marketing techniques (mentioned earlier) to do so. Probably 50 000+ potential users exist as of today, and if we do a good job promoting, I think we could have a good percentage of them using our application within a year or so. That would generate a good ad revenue to keep us going.

5 Reflection on Project Management (LO24)

[This question focuses on processes and tools used for project management. —TPLT]

5.1 How Does Your Project Management Compare to Your Development Plan

[Did you follow your Development plan, with respect to the team meeting plan, team communication plan, team member roles and workflow plan. Did you use the technology you planned on using? —TPLT]

5.2 What Went Well?

[What went well for your project management in terms of processes and technology? —TPLT]

5.3 What Went Wrong?

To start in terms of communication plan, plan was to meet every sunday and tuesday for 2 hours, but with a lot of us having 5 other courses and personal life outside of school, there were a few occasions where these dates could not be fulfilled and when trying to find an alternate date it was sometimes difficult to set a new time. Also for a lot of the assignments/tasks we tend to set soft deadlines and strict deadlines, but what we noticed was that usually majority of the team did not meet the soft deadlines and usually followed the strict deadlines. Finally a lot of the communication/project management was done via Discord, Facebook Messenger, Git Issues and Gantt Chart, these tools were effective but sometimes the organization of tasks were not the best, so maybe

using a tool such as Jira, where they show tasks via an organized table could have been more effective.

5.4 What Would you Do Differently Next Time?

To start, for communication plan every meeting was sunday and tuesday for 2 hours but should have also considered alternate dates such monday for example if team members have conflicts, in order to make meetings more consistent if a conflict arises. Also, as mentioned in what went wrong, team members tended to ignore soft deadlines and tended to only follow strict deadlines maybe should have implemented a mechanism to ensure more soft deadlines were met, such as maybe trying to divide the work into even smaller chunks and have only strict deadlines. Another thing is would have used more tools that have better visuals and improve organization for tasks such as Jira, so it is easier to keep track of work done. Also in the team meeting we did take notes of each meeting, but we could have assigned one note taker per meeting instead of all of us doing it at the same time.