Implementation Documentation

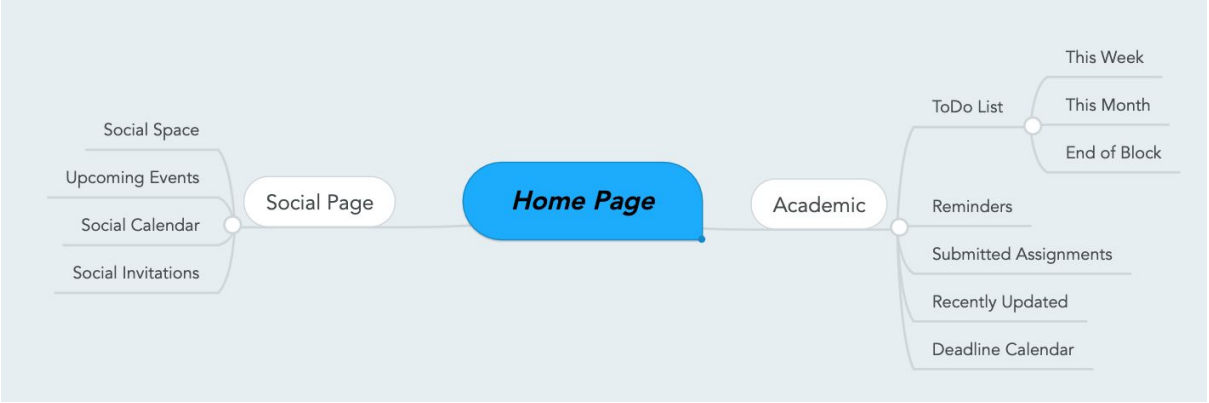
Mission

We are going to implement a new version of the VU app focused on improving the deadline management and social awareness amongst students.

Technology of the Prototype

Keeping in mind the short period of time we are given to develop this prototype and the commonly growing unwillingness of people to download mobile applications, we decided to build a dynamic website optimized for mobile devices. This will be an advantage making it openly available to all students without the need to install an app. Furthermore, we are familiar with the web technologies which gave us more confidence in developing a satisfying prototype representation of the real app as envisioned in the project ideation.

Below is the illustration representing the basic app structure.



Fully Functional Parts

Based on our knowledge of the technology in which we choose to develop the prototype and the limited amount of time we have decided to focus on the decisive parts of the app that provide a dynamic and an adequate representation of the final product.
These parts include:

- Todo List *with three filtering options (by week, month, or end of block)
- List of recently updated assignments
- List of submitted assignments
- Upcoming Events Feed
- Social Calendar
- Social Space (Invitations from and to other students)

Not Implemented Parts

There are a few features that we envisioned to be present in the final product but have decided to omit in this prototype implementation. This decision has been made based either on the limited time given or the complexity of the feature in question.

Parts that will be omitted as well as their workarounds are described below.

1. Sending Social Invitations
    a. In order to properly represent the social space of the app, the user in this working prototype will be able to follow the process of sending a social invitation, however, mostly due to the lack of an existing database of contacts, there will be no response from the server on the completion of the request.
2. Completing the Assignment
    a. We acknowledge that submitting an assignment is an essential part of the prototype, thus, the user has the option to follow that process. However, due to the lack of knowledge in the file transfer process and the lack of the proper infrastructure for the app's database, the option to submit a file in conjunction with the assignment is unavailable. The prototype will however register that the assignment has been submitted, and no longer display it in the toDo list, and instead in the submitted list.
3. Setting a Reminder
    a. Lastly, the user will have access to a screen where he/she would normally set a reminder. We decided to supplement this page with a static values representation of the final feature. This decision was made based on the premise that the web technology used to build the prototype is limited in its ability to interact with the mobile phone settings. Thus, the setting of a reminder would be difficult, as a working website has no permission to send reminders to a device unless currently open.

Plan for Implementation

Before we began the implementation we decided to divide the project based on the skills we previously acquired inorder to work most efficiently and be able to develop a working prototype. Therefore, Leander is going to be responsible for developing the PHP code that will handle requests and communication with the database. Franco and Oskar are responsible for building the HTML and CSS framework of the app and making it visually appealing.

Date	Leander	Oskar	Franco
10th June			Set up the file structure of the system and create all relevant HTML/PHP files of the corresponding social

			and academic pages.
11th June	Create the database with two tables: assignment and VUEvents with corresponding attributes that enable interaction with the user.		
12th June	Connect the social pages to the VUEvent table, so that the user can interact with the Events.		
13th June	Connect the academic pages to the assignment table, so that the user can interact with the toDo lists.	Add CSS to the social pages of the system, so that the user can properly interact with the system and has a positive experience.	
14th June		Add CSS to the academic pages of the system, so that the user can properly interact with the system and has a positive experience.	
15th June	Navigate through the app and see if everythings works. Take care of bugs and make a list of improvements.	Navigate through the app and see if everythings works. Take care of bugs and make a list of improvements.	Navigate through the app and see if everythings works. Take care of bugs and implement improvements.
16th June			Implement a list of improvements and hand in assignment.

Backup Plan

In case we deviate from the aforementioned plan the backup plan is to substitute the failed implementations with static values. This will be acceptable in the case of our system because its prototype static counterpart is going to be representative of the features people are familiar with from a range of other applications.

Handling Failed Implementation

We had the advantage of being familiar with the programming language we choose to develop the prototype in, therefore, we decided that if there were any deviations from the plan due to features we are unable to implement that we would use our knowledge of the language and technology to find a static solution that would simulate the functionality of that particular feature.