UADY

UNIVERSIDAD
AUTÓNOMA
DE YUCATÁN

Software Engineering

-Software Engineering Fundamentals-

Second Increment

Osiris Team Members:
Fernan Enrique Cetina Escalante
Jorge Teodoro Dawn Rodriguez
Rodrigo Alejandro Castrejón Cervantes
Cinthia January Huchin Pedrero
Ricardo Reyes Balam Cupul

Mérida, Yucatán. January 9th, 2021.

Functional requirements and not functional requirements (Final version):

- The system shall provide a short introductory tutorial that teaches users its main features: add, edit and remove tasks, update progress, and change notification frequency.
    - It should be accessible via a separate tab at any time.
- The user shall be able to see a list of all of their current active tasks. It is very important to display due date and progress status.
    - The tasks should be organized by their due date, nearest one first.
- The user shall be able to add, edit or remove a task from their task list.
    - These actions should appear in the list view and should take one click to use them.
    - Our system should not save any identifiable information from users, these being: name, username, and email.
- The user shall be able to set a due date when creating a task and also be able to modify it [due date] via the button containing the date.
    - The due date should be set using a date picker.
- The system should show a menu to help users reassign a task when a user misses its due date.
    - A calendar should be displaced to pick a new date.
    - The system should allow us to remove or reschedule the task.
- The system shall display the current progress of each task and allow the user to update said progress.
    - The system should allow for progress to be reverted in case of a revision or an accidental update.
    - The ‘progress update’ action should be accessible directly on the task list, right next to each task.
    - It should only take one click to update the progress.
- When the user has active tasks, the system shall keep pushing notifications saying “You have pending duties” until they complete them.
    - By default, the system should only send one notification per day (10:00 hrs).
    - The user shall be able to set the notification frequency.
- Each task shall allow users to append ‘update notes’ to them in order to keep better progress.
    - The notes should be displayed alongside its corresponding task in the task list.
    - Notes should be autosaved.

Modeling

We used figma.com for the Wireframe(‘mock up’) modeling. Here we represented the UI and saved all traceability aspects to ensure high quality on static and some dynamic testing.
It can be seen here: https://bit.ly/3rizTGe
The main goal of Team Osiris is to provide itself or another team of software development enough tools to guide their own project with the specified requirements. As such it is not displayed: Architectural, Data Types and Procedural design, because it is not intended to show that level of specificity, each team can decide what fits best on future technologies.

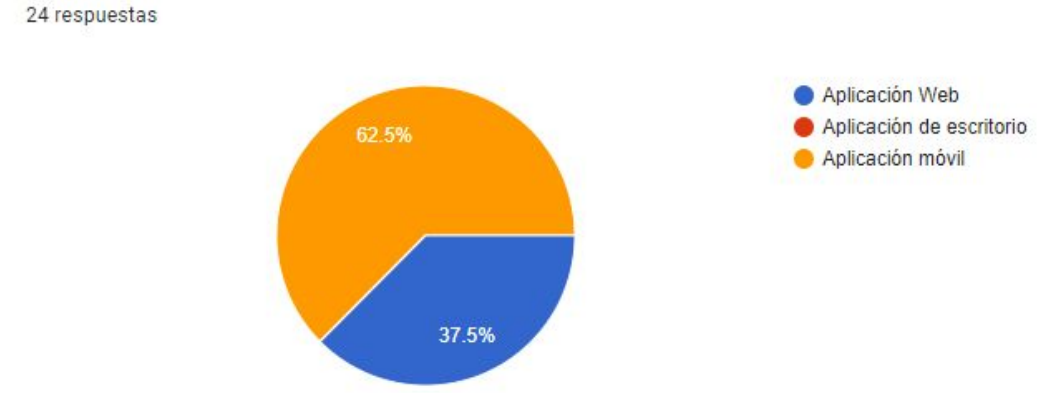
Implementation

For the platform of Osiris we presented the customers with a new survey, in which 24 participants were registered and decided ‘Mobile App’ for the platform; but for this early stage of Osiris, ‘WebApp’ can be really helpful to make instant updates and guide the UI and back/front end for the Mobile App, that's why we will make functional prototypes of the product on Web but on further stages it will be deployed con Android & IOS.

For the implementation phase we considered generating the functional code with the JavaScript framework React, we use this framework for the functionalities and facilities that it provides. In this way we focus on developing prototypes of functional interfaces for the users and start evaluating how the user would interact with our system. The implementation follows a continuous integration methodology where we create the root component and each member works with separate components and merges them to the system, all this process is made in our GitHub repository.

For considerations of the schedule we aren’t making the commitment of releasing the code for this increment, just in case that we could finish the MVP of the code generation it would be presented in the final delivery.

(Results of the survey)



Translation: 24 answers. WebApp(blue); Mobile App(yellow); Desktop App (red)

Testing

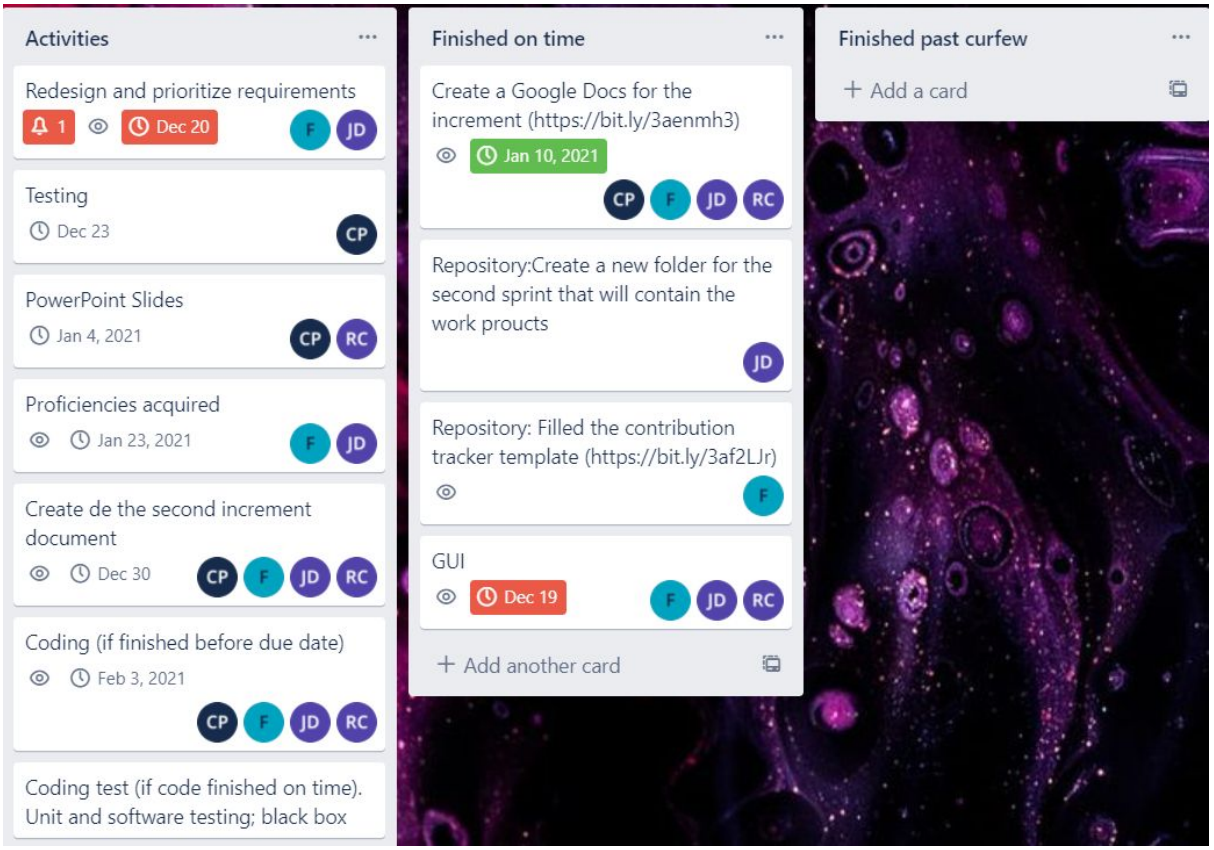
For the project three types of tests should be carried out. The first one being inspection, where the Team with the Mentor will decide if each individual requirement is properly represented on the design. The second test is ‘acceptance testing’ where 6 random students will be shown the Wireframe and later asked relevant questions that are intended to help the development of a new version of the project and successfully have validation from users. And the third type of testing will be a ‘5 seconds test’, information will be gathered and imported to a spreadsheet, this is intended to guide the UI design.

Mentor revisión:

Acceptance testing (Q/A):

5 seconds test (Q/A):

Project Planning / Teamwork:



We used Trello for the assignment of the duties, the accountables(Fernan Cetina[F], Jorge Rodríguez[JD], Rodrigo Castrejón[RC] y Cinthia Huchin[CP]) and due dates. The artifacts with accountables are self explanatory on the Trello but as a list they are:

1. the UI model (F, JD, RC)
2. this document (Team Osiris)
3. PPT presentation (CP, RC)
4. The contribution tracker (F, JD)
5. Github binnacles and project advances (Team Osiris)

It can also be seen that a Coding phase is planned but not mandatory for the MVP of the increment.

Furthermore we have 3 lists according to the First Increment document methodology (activities, finished on time and finished past curfew).

Repository: https://github.com/FernanCetinaE/TeamOsiris

For the team dynamic and process it is used ‘Scrum Methodology’ and can be reviewed at length on the First Increment Document available in the Repository.

As for the format of the contribution tracker we implemented this scheme:

Sprint: <increment>						
Accountables	Commitments bases on activities		Times the project was delaied by needing corrections or somebody else did your commitment	Team mettings attended and mettings with the mentor	On 100% scale	Final grade
	Total	Completed on time				
Fernan Enrique Cetina Escalante					%	%
Jorge Teodoro Dawn Rodriguez					%	%
Rodrigo Alejandro Castrejón Cervantes					%	%
Cinthia January Huchin Pedrero					%	%
Ricardo Reyes Balam Cupul					%	%

And we used it as follows:

Contribution tracker					
Sprint: <increment>					
Accountables	Commitments bases on activities		Times the project was delaied by needing corrections or somebody else did your commitment	Team mettings attended and mettings with the mentor	On 100% scale
	Total	Completed on time			
Fernan Enrique Cetina Escalante	7	3/3		1	%
Jorge Teodoro Dawn Rodriguez	7	3/3		1	%
Rodrigo Alejandro Castrejón Cervantes	5	2/2		1	%
Cinthia January Huchin Pedrero	5	1/1		1	%
Ricardo Reyes Balam Cupul	0	No aplica		0	%

Contribution calculus:

[(Activities completed in time/Total activities + your attendance/highest attendance)/2 - if(delay>3){delay*5%} else {without sanction}] * 20%

At the end of the sprint, each memeber is expected to contribute 20% in total

Subject Proficiencies: