PROJECT PLAN 1.0

Share it!
UAM

BY P.D. TEAM

Juan Fernández del Pozo Romero
Guillermo Martin-Coello Juárez
Rodrigo García Redero
Daniel Varela Sánchez

Madrid, 20/04/22

Reference: EcoUAM-ShareIt/Project Plan/v.1

Version History Table

Version	Date	Contents or main modifications	Purpose
0.1	10/04/2022	Initial division	Start document
0.2	11/04/2022	Introduction	Introduction creation
0.3	12/04/2022	Project overview	Project overview creation
0.4	13/04/2022	Solution to Employ	Solution to Employ creation
0.5	14/04/2022	Project Management	Project Management creation
0.6	15/04/2022	Monitoring and Control	Monitoring and Control creation
0.7	16/04/2022	Conclusions	Conclusions creation
0.8	17/04/2022	Appendices	Appendices creation
0.9	18/04/2022	Summary	Summary creation
1.0	18/04/2022	Error checking, final revision	Validation of project

Summary

After completing the Technical Report about Share It!, proposed for the EcoUAM competition for IT Innovation Projects, the project is selected and new details have to be described.

This document goes over the new requirements that the project will now include, such as two new subsystems for rating and managing incidents. The former requirements are reformulated after validating them with the client. Another main goal of the document is to provide an approach on how the design will be, the tools that it will use, and methodologies that will be applied to it. Furthermore, the monitorization and control of those methodologies is described.

The document elaborates on a project plan, considering the Function Points estimation technique. The different subsystems are measured by means of effort required and complexity, and this information is considered along with our capabilities and resources. We obtained and estimated duration of about one year (374 days of work), employing a fully qualified team of 6 people. This planning is the result of an optimization process provided by the MS-Project tool.

The deliveries will be incremental during the development of the project, so the client is able to validate and try out the different aspects of the application. It has been taken into account that the project is new on its domain and might require modified or new requirements during the development.

Finally, the document provides an estimation of the costs that the project will suppose. Taking into account the development team, the infrastructure, and the new required equipment: the analysts provided a total cost of around 317.735,00€.

Table of Contents

1. Introduction 1
1.1 Purpose 1
1.2 Scope 2
1.3 Responsibilities 2
1.4 Definitions 2
1.5 Referenced Documentation 3
2. Project Overview 4
2.1 Project Description 4
2.1.1 Goals of the project 4
2.1.2 Subsystems 4
2.1.3 Scope of the project 4
2.2 Requirements 5
2.2.1 Functional requirements 5
2.2.2 Non Functional requirements 14
2.3 Deliverables 15
3. Solution to Employ 16
ARCHITECTURE 16
PORTABILITY 16
4. Project Management 17
4.1 Estimations of the Software System 17
4.1.1 Function Points 17
4.1.2 Complexity factors 20
4.1.3 Grouped Summary 21
4.2 Organizational Structure 21
4.3 Assigned Qualified Personnel 21
4.4 Time Management 22
4.5 Costs Management 26
4.6 Quality Management 27
4.7 Risks Management 27
4.8 Acquisitions Management 28
4.9 Documentation Management 28
5. Monitoring and Control 29
5.1 Configuration Management 29
5.2 Progress Monitorization 30
5.3 Verifications at each Phase 30
5.4 Testing and Validation 31
6. Conclusions 32

APPENDICES 33
Appendix A: Function Points. 33
Users Management Subsystem 33
User Assessment Subsystem 35
Publication of Rides Subsystem 39
Ride Seat Booking Management Subsystem 42
Payment Management Subsystem 47
Incidents Management Subsystem 50
Appendix B: Complexity Factors. 53
Appendix C: Assigned Qualified Personnel 56
.... 58
Appendix D: Meetings 59

1. Introduction

1.1 Purpose

The purpose of this document is to perform the planning activities for the Project aimed at the Development of a Software System for Vehicle Sharing at the Autonomous University of Madrid (UAM) by the team "Pings development team". For which the application "Share it!" would be developed.

The development of this project follows an incremental and iterative life cycle model, with three increments, each with the same phases (requirements analysis, design, coding, unit tests, integration tests and deployment).

In section 1 the project's scope is described, the responsibilities of the project, some definitions and acronyms, and referenced documentation.

Section 2 contains a project overview including its problems, as well as the subsystems and requirements of each of them. Finally, the deliverables to be handled are listed.

Section 3 includes the system's architecture and methodology are described.

In section 4 a detailed estimation and planification of the project is described. Estimation using Function Points Estimation Technique, and also the Microsoft Project tool, to help in all the estimations.

In section 5, the process monitorization of how the project evolves, as well as the features of the verification of each phase.

Section 6 includes the conclusions.

1.2 Scope

Section 2 describes the requirements and subsystems that will be provided to the client. As a result, the client can make new modifications or deletions from the document's specifications. However, throughout the analysis process, the criteria might be changed. With this information, an estimate of the project's cost and time would be calculated.

1.3 Responsibilities

PROJECT LEADER	Juan Fernández del Pozo Romero
PROJECT HEAD	Guillermo Martin-Coello Juarez
PROJECT QUALITY MANAGER	Rodrigo García Redero
PROJECT DOCUMENTATION MANAGER	Daniel Varela Sánchez
CUSTOMER REPRESENTATIVE	Daniel Varela Sánchez
OTHER	None

1.4 Definitions

FR	Functional requirement
NFR	Nonfunctional requirement
ILF	Internal logical file
EIF	External interface files
EI	External input
EO	External output
EQ	External inquiry or query
DET	Data element type

FTR	File type references
MS-Project	Microsoft Project
IOS	iPhone Operating System
DB	Database
FAQ	Frequently Asked Question(s)
GUI	Graphical user interface
POS	Point of sale; point of service
API	Application programming interface
SDK	Software development kit
UAM	Universidad autónoma de madrid
UM	User management
RSB	Ride seat booking
PR	Publication of rides
IM	Incidents management
PM	Payment management
UA	User assessment

1.5 Referenced Documentation

REFERENCE	TITLE
1	Team-Quality-Management-Document
2	Team-Risk-Management-Document
3	Team-Configuration-Management-Document

2. Project Overview

2.1 Project Description

This section is divided into three subparts: The goal of the project, the scope of the project and the subsystem that lets the project accomplish the goals.

2.1.1 Goals of the project

The objective of the project is to provide the teachers and students from UAM a fast way car share rides to the university. This will help obtain other goals such as reducing car usage, traffic and contamination around UAM, while also making the rides to it more economical to the users.

To accomplish these goals a software application must be designed with different subsystem that together will build an application for the user and provide the expected experience.

2.1.2 Subsystems

The project is divided in different subsystems to approach each objective of the application functionality separately and provide a more modular and easy to manage product. These subsystems are the following ones:

- User Management subsystem: the subsystem will provide the functionality related to the user account such as login into the application, login out of the application, seeing user profiles or sending messages between user accounts.
- User Assessment subsystem: the subsystem will provide the functionality related to user rating, such as rating passengers and drivers, making assessments and observations, blocking accounts if ratings are too low or unblocking if desired.
- Ride seat booking management subsystem: the subsystem will provide the functionality related to see, filter, search and book rides. It will also check seat occupation, calculate prices and show notifications related to the booking.
- Publication of rides subsystem: the subsystem will provide the functionality related to the publication of rides by the drivers to the platform. This includes fill all the expected parameters of the ride like the number of seats and price, and the possibility of modifying or canceling rides.
- Payment management subsystem: the subsystem will provide the functionality related to the payments on the application. This includes conversion of money to credits, transference of credits and credits wallet management.
- Incidents management subsystem: the subsystem will provide the functionality related to incident reports by both drivers and passengers, incidents status, updates and solutions.

2.1.3 Scope of the project

The software proposed is aiming to provide teachers and students of UAM a viable alternative to obtain a cheap and fast way of travel. Car sharing already exists for this purpose, but the application will provide a more visible and easier more structured way of doing it.

2.2 Requirements

2.2.1 Functional requirements

2.2.1.1 User Management - UM

- **FR-UM-1:** Only authenticated users will be able to access the application (except for the login page). When users log in, the system will verify the access data (UAM email and password) entered by them with the UAM authentication system.
    A. If the user is not logged in, he will see a login page.
    B. To access the page users, have to write username and password in the login interface.
    C. When the user writes username, password and other data fields correctly and clicks the login button the application functionality will become available.
    D. Data written in the login interface (username and password) will be processed by the user authentication system.
    E. If the authentication is accepted, the system will assign a session to the user (Authenticated user).
    F. If the UAM authentication system returns a valid response a new user session will be created, and the application will be shown to the user.
    G. If the authentication system returns a not valid response the login interface will show again with an Invalid Username or Password message.
- **FR-UM-2:** When users log out, the system will delete their session (unauthenticated user).
    A. To log out, the user must click on the settings button and then on the logout button. The user session will then end. The system will save all user data in case something has changed and then it will delete the current session.
    B. The application will stop being available.
    C. The login interface will show up for a new user to login.
- **FR-UM-3:** When users access the system for the first time, an "empty" profile will be created (containing only email and password).
    A. If the user has not used the application before, a new user profile with the user data will be created when the login button is pressed.
    B. A new virtual payment account will be created by the system with basic payment information of the user.
- **FR-UM-4:** Authenticated users will be able to see the rides published and reserved by them (historical) as well as the payments they have made for each ride.
    A. Authenticated users can see a list with all published and reserved rides (by the current user) by pressing on the list button in the main page and then my rides button. Rides will be organized on a historical order on the list. Payments of each ride will be shown next to each ride data.

- **FR-UM-5:** Authenticated users will be able to see the profile of the user offering a ride.
    A. Authenticated users can select the user icon of a specific ride to see the user profile of the driver. This will display a profile page with all the information about the drive, including name, age, picture and reviews.

- **FR-UM-6:** Authenticated users will be able to send and receive messages
    A. The users will have a chat to interact between each other.
    B. Users will see if the chat is for passenger or driver mode.
        a. Passengers will be able to speak with drivers of booked and confirmed trips.
        b. Drivers will see passenger chats for paid trips. Additionally, they will also be able to chat with passengers that have only booked a seat.

2.2.1.2 User Assessment - UA

- **FR-UA-1:** The driver of a ride must be able to rate the passengers of his rides from one to five (lowest to highest respectively)
    A. Driver must be logged in.
    B. The user has to select the ride menu where the user to be reviewed is.
    C. The user has to select the user to review.
    D. The ride must have been finished.
- **FR-UA-2:** The driver must be able to specify the qualities that the passengers had from a quality list (for example, talkative, reserved, timely, etc.) And also, will have an open field for additional observations on the passenger.
    A. The driver must select the ride.
    B. The driver must select the passenger to review.
    C. Options with qualities to be assigned to the reviewed user will be displayed.
    D. A textbox field will be displayed for additional observations.
- **FR-UA-3:** The passengers must also be able to make assessments and observations to the other passengers of the shared ride.
    A. The user must select the ride.
    B. The user must select the passenger to review.
    C. Options with qualities to be assigned to the reviewed user will be displayed.
- **FR-UA-4:** The system will block a user's account if they receive a big number of really poor scores.
    A. If a big number of poor scores is received, the user account will be blocked.
    B. The user will not be able to access the blocked account (log in).
    C. If login is attempted, a message will be displayed showing the account is blocked.
- **FR-UA-5:** The users must be able to file a claim for the account blocked if they think the app has made a mistake and their case will be reviewed
    A. If login is attempted and the user account is blocked a file claim button will be displayed.

B. If the file claim button is pressed, the user will be able to write the claim.
C. If the file submit button is pressed, a claim will be made to the administrator.

- **FR-UA-6:** The users should be notified when the claim on the account blocking has been reviewed
  A. The user must have filed a claim.
  B. If the user logs in and the claim has been denied a message will display showing the claim was denied.
  C. If the user logs in and the claim has been accepted the user will be able to login.
  D. An email will automatically be sent to the user when the account blocking review has been done.
- **FR-UA-7:** The administrator will receive account unblock claims and will be able to accept them or deny them.
  A. A notification will be shown to the administrator when a claim is filed.
  B. The administrator can see all the filed claims and their corresponding users.
  C. The administrator can press on the claim's user to see their poor reviews.
  D. The administrator can press an "accept claim" button to accept each claim or "deny claim" button to deny it.

2.2.1.3 Ride Seat Booking Management - RSB

- **FR-RSB-1:** Users will be able to see the available rides.
  A. The system will show a button for the user to interact and will display all available rides.
  B. The available rides button will show on the main screen.
  C. For users to see the rides they will need to be authenticated.
  - **FR-RSB-1.1:** To see the rides passengers must indicate the pick-up point, date and time, and the arrival point.
    A. In the rides display screen a filter interaction system to find rides, with a field to choose date, time origin and arrival.
    B. To change the filters the user will need to press the filter screen.
    C. Once these filters are used the system will only display rides that satisfy the search.
  - **FR-RSB-1.2:** In addition, they will be able to define filters for ride details
    A. The system will also show filters for details such as smoke and car type.
    B. Users need to be logged in and have accessed the rides page.
    C. User will use search filters to change pick-up point, date and time, and the arrival point. At the same time the user can modify these details filters.
    D. The application searches for the rides that satisfy the conditions.
  - **FR-RSB-1.3:** When filters are applied the system will display all the results that match the specified criteria.
    A. Users need to be authenticated.
    B. Application search for rides that satisfy the search.
    C. Users see the available rides displayed on the screen with the corresponding drivers dates and times.

- **FR-RSB-2:** Passengers will be able to see the ride details.
    A. When the user is authenticated, the system will display the option to see the ride detail of a specific ride the user selects on the rides menu.
    B. The user needs to be authenticated.
    C. The user has to enter the rides menu. The system will display the available rides.
    D. The user must press a ride to open the rides detail menu.
    E. The system will access the database and show all the ride information to the user.

- **FR-RSB-3:** Passengers can make seat bookings when they are seeing the ride details.
    A. If a user is on a ride detail screen a button can be pressed to book a seat on the desired car. The system will add the passenger to the respective ride.
    - **FR-RSB-3.1:** For recurring journeys, the system will require passengers to confirm if the reservation is for a particular ride or it is intended to be a recurring booking (in this case the final date must be provided).
        A. When the book seat button is pressed by the user a pop-up message will be shown asking if the ride is recurrent. If so, the system will repeat the process for all the recurrent rides of that trip.
    - **FR-RSB-3.2:** The system will calculate and show the user the total cost of the book.
        A. A user that has been previously authenticated
        B. The user is in the process of booking a ride.
        C. The system will calculate the total cost of the booking and display it to the user.
    - **FR-RSB-3.3:** The system will verify that the user has the required credit in his/her virtual payment account.
        A. The system will access the database to find the user balance and check if the user booking has enough credits to fulfill the book cost.
        B. If the user has enough credits to pay the book, the system will change the user balance account in the database, where the cost in credits will change from available credit to blocked credit.
        C. After making a successful booking by a user, the system will show a reservation request on the request menu of the driver client of the application.
        D. If a reservation request is done, it will show to the driver application.
            a. A request for a ride has been made by a user.
            b. The system shows the rider a new request popup on the request menu icon in the main page.
            c. The driver enters the request menu, and the system shows all reservation requests pending for the driver.
            d. The driver accepts the request. The system adds the user to the corresponding rides in the database.
            e. The system generates a booking voucher with a unique code and sends it to the passenger.
        E. If the driver does not accept, the passenger credit that has been blocked for that ride will be unblocked.
            a. If a reservation request is done, it will show to the driver application.

b. A request for a ride has been made by a user.
c. The system shows the rider a new request popup on the request menu icon in the main page.
d. The driver enters the request menu, and the system shows all reservation requests pending for the driver.
e. The driver rejects the request. The system removes the request and unblocks the previously blocked money of the passenger in the database.

F. If there are not enough credits, the user cannot book the ride(s)
a. The user is authenticated
b. The user accesses a ride detail page
c. The user tries to book a ride with the book ride button.
d. The system checks in the database if there is enough credit.
e. If there is not the system would not allow the booking. Feedback of the cancellation of the process is shown in the user interface.

- **FR-RSB-3.4:** The system will indicate the occupation of the booked seat(s) in the ride.
  A. When a user sees a ride detail screen of a specific ride, the occupation of the seats will be shown.
  B. When a book is validated, the system will add the user to the ride.
  C. The system will check for the number of users of a ride on the database and will display them showing some visual feedback on how many seats are left on the ride detail page.

- **FR-RSB-3.5:** Passengers cannot reserve seats on different rides that overlap in time.

A. A passenger reserves a seat on one ride at a specific hour.
B. The passenger tries to reserve another seat on another ride in that same hour.
C. The system cancels the operation and shows to the user some feedback response indicating why the procedure could not be achieved.

- **FR-RSB-4:** Passengers may cancel any reservations made before the beginning of the ride. A passenger can press the cancel button on the details of my rides page before the beginning of the ride.
  A. The user accesses my rides button in the main page.
  B. The system shows all current rides booked or done by the user.
  C. The user selects the ride desired to cancel. The system displays the ride details with a cancel button. The user can press the cancel button.
  D. The system will access the database to change the user balance and the available seats.
  E. If the cancellation has been made within 24-3 hours before the ride a 30% of the price would be not returned.
  F. If the cancellation has been made within 3-0 hours before the ride there would not be a refund.

- **FR-RSB-6:** Passengers will be able to check the rides they have reserved (both those from the past and the ones for the future)
    A. User is authenticated.
    B. The user presses my rides button in the main page.
    C. The system displays past history of rides. A button is also displayed to change to the future history rides.
    D. The system will display an icon next to each future ride showing the state of the book (pending, accepted or rejected).
- **FR-RSB-7:** Riders will receive notifications related to bookings when the driver edits the ride.
    A. Drivers can change the details of the rides or cancel the ride. For changes that affect the ride such as changing the date the users will get a notification.
        a. If the ride is canceled by the driver, the system will automatically refund the passenger's credits and emit a notification.
    B. For recurring journeys, passengers will be notified before rides that a scheduled ride is approaching.
    C. When the driver accepts the booking request.
- **FR-RSB-8:** Drivers will receive notifications related to bookings when the passenger books a ride.
    A. When a request for a ride has been made by a user.

2.2.1.4 Publication of Rides - PR

- **FR-PR-1:** Authenticated users will be able to publish rides for carpooling.
    A. In the driver's main page there will be a button to create a ride once the button is pressed the ride creation page will be displayed.
    - **FR-PR-1.1:** They must indicate the place of departure, date and time, number of persons admitted, whether smoking is allowed in the vehicle, destination, and price per seat. They can also include (optional) a textual description with pick-up points along the ride and/or type of vehicle.
        A. The ride creation page will contain all the fields needed to create the ride.
        B. Once the user fills in all the ride details, the publish ride button will activate and he will be able to publish his ride.
        C. The ride then will get stored on the external rides database.
    - **FR-PR-1.2:** Optionally, users can indicate whether it is an occasional or recurring ride.
        A. For frequent rides drivers must indicate how often they take them and a date when they estimate to quit taking this ride (maximum 2 months).
            a. Once the ride is marked as recurring, a button for each day of the week will be shown and the user will be able to select the days of the week he is going to make this ride (we assume that the recurrence of the rides are weekly and that if there are some specific days the user will not make the ride he will cancel those days manually later).

        b. Also, when marked as recurring, a date will be asked to the user indicating when he will stop making the ride.

    B. The system will register all the rides configured by the user (a single ride for sporadic rides, and all possible rides for recurring rides).

        a. The application will store in the database the ride and will later allow the user to see the rides that he has taken and published.

- **FR-PR-2:** Drivers will be able to change the number of available seats if it does not affect a reserved seat (that is, a seat that has been reserved cannot be removed).
    A. In the driver main page, he will have to press the list button, then the my rides button.
    B. In my rides page you can access the main details of each ride. When a ride is pressed the ride's complete details are shown, and if the ride is published by that user, he can change the details there and then press the button to save changes.
    C. If the changes are valid, a page showing that the changes have been made shows, but if they are not, a n error page is shown.
- **FR-PR-3:** Drivers will be able to modify any information of the ride if no reservation has been made for it. When recurring rides are modified, drivers must indicate whether the modification affects just one specific ride or all rides from the given date on.
    A. Following the steps on the previous description the driver can access the ride modification page where if no passenger has reserved his spot, the driver can make any change he wants.
    B. Once the change is made, if the ride is recurring, a pop up will show that will ask the driver if the change should be applied to all the rides or just the single one edited.
- **FR-PR-4:** Drivers can cancel rides (they can be completely canceled or, in case they are recurring, just for one day).
    A. The user would have to enter in my rides page, then he would have to enter the details of a ride that he has published, from there a cancel ride button will show, and when it is pressed on a recurring ride, a message asking to apply the cancelation for one ride or for all will show.
    - **FR-PR-4.1:** When a ride is canceled, the blocked credits of the passengers are refunded.
        A. Once the ride is canceled by the driver, the system will automatically refund the passenger's credits and the ride will appear on the system as canceled and never done.
- **FR-PR-5:** Drivers will be able to see the rides they have published (both those already made and those that are still pending)
    A. In the driver main page, he can press the list button that will take him to his rides page.
    B. In that page he will find two tabs, past and future rides. By pressing on the one he wants to see he will be able to visualize the past or the future rides he has been on, will be on or published.
    - **FR-PR-5.1:** Drivers will be able to enter the code given by passengers on the rides made that have pending payments. If the code entered is correct, the system confirms the journey made by the driver and the passenger (it initiates the transfer of passenger-driver credits, see Payment Management environment).

A. Once the ride time is past, the passengers will be able to see a pop up with a code once they access their current ride.
B. Once the time of the ride is past, the driver will have an option to input the passengers' codes to confirm their ride. From the details of the current ride accessed in the rides page, he will find an option that says scan confirmation codes that will allow him to input the passengers' codes in the text field and the system will confirm the payment from the passenger to the driver.

2.2.1.5 Payment Management - PM

- **FR-PM-1:** Registered users will have a virtual payment account (VPA) associated with their identity.
  A. Registered Users have a profile section with the VPA settings menu
  B. The sub-section has a form for non-configured accounts
    a. The form asks for the credit card number, expiration date, card pin
    b. The user has to validate its identity
    c. The card remains set for future payments in time
  C. Accounts with already configured VPA will have an option to add more cards
  - **FR-PM-1.1:** VPAs will initially have a balance of 0 credits. One VPA credit is equivalent to €1. Credits can be divided into cents.
    A. The User has a balance section on his profile
    B. The current credit is displayed in units and cents
- **FR-PM-2:** VPAs will manage available credits
  A. Available credits are the number of credits that the user can use at a given moment.
  B. Blocked credits are user's credits reserved by the system as a guarantee of
  C. payment for booked rides. Blocked credits cannot be used by the user unless they are unlocked.
  D. Total credits are the total amount of credits associated with the account at a given moment (the sum of available + blocked credits).
- **FR-PM-3:** Authenticated users can transfer credits to their VPA from their bank account through an operation by means of virtual POS (online payment).
  A. The user has a "Deposit" option on its balance section.
  B. The user enters the desired amount to add to the account.
  C. The system proceeds with the POS tool.
  D. Transferred credits will be added to the available ones.
    a. The new credit is computed and displayed.
- **FR-PM-4:** Users will be able to request the reimbursement of all or part of their "available" credits in their VPA to a bank account.
  A. The user has a "Withdraw" option on its balance section.
  B. The user enters the desired amount to subtract from the account.
  C. The system proceeds with the reimbursement module.
  D. In case of success, the system will deduct the reimbursed amount of available VPA credits.

a. The new credit is computed and displayed.

- **FR-PM-5:** When a driver confirms that he/she has made a ride with a passenger, the transferred credits will be added as available credits of the driver and will be deducted from the blocked credits of the passenger.
  A. The driver confirms a drive with a specific passenger is over.
  B. The system checks that the passenger has not issued any incidents during the trip.
  C. Credits blocked from the passenger VPA are transferred to the drivers VPA.
  D. New balances are computed in both accounts.
  - **FR-PM-5.1:** When a ride is canceled, the corresponding blocked credits are refunded to its respective owners (unblocking them, subtracting them from the blocked credits and adding them to the available credits).
    A. The driver cancels a ride
      a. Some passenger cancels a certain ride
      b. The passenger specifies the reasons to cancel the ride
    B. The credits are unblocked and remain in the passengers VPA
    C. The new balance is re-computed

2.2.1.6 Incidents Management - IM

- **FR-IM-1:** An incident can be reported by any rider (driver or passenger).
  A. A ride was published in the system by a rider.
  B. A user signs to be a passenger of that ride.
  C. Any of them reports an incident of a category with a description.

- **FR-IM-2:** A registration of an incident will block the credits associated with the trip.
  A. New incidents reported by any ride will block the credits related to the ride.
  B. No transaction will be done until the manager's decision is made.

- **FR-IM-3:** Incidents will have a status (issued, in revision, solved), a category, and a description of the problem that happened.
  A. Category is general.
  B. Status can only be changed by the manager.
  C. Description will show a detailed view of the incident.

- **FR-IM-4:** Updates in an incident status will be received by the users as notifications.
  A. The incident is issued
  B. The incident is under revision from the manager
  C. A decision for the incident has been made by the manager.

- **FR-IM-5** Solved incidents will arrange the credits of the ride based on the manager's decision.
  A. The manager updates the incident with a decision.
  B. If the decision benefits the driver the payment resolves as usual.

    C. If the decision benefits the passenger, some of the credits will be unblocked and returned to his VPA.

  - **FR-IM-5.1:** Drivers may face a penalty in the form of a reduction of credits he should obtain for the journey.
    A. The driver receives a reduced percentage of the original fee based on the severity of the incident.

2.2.2 Non-Functional requirements

Interface and usability requirements

- **NFR-1:** The background of the application will be white.
- **NFR-2:** The user will move through the application functionalities by the use of buttons.
- **NFR-3:** Important text and buttons will be displayed in green, following a close esthetic to the UAM.
- **NFR-4:** Less important text will be displayed black.

Operational requirements

- **NFR-5:** The system must have a daily backup system.
- **NFR-6:** If a transaction is not successful, restore functions will be available.

Documentation requirements

- **NFR-7:** Application will only be available in English, as it is the language of the client.
- **NFR-8:** User types will be clients and a unique administrator.
- **NFR-9:** social media of the company will be available for contact.
- **NFR-10:** FAQ section will be available in the app
- **NFR-11:** Application policy will be available in the app
- **NFR-12:** Explanation on how to use the software will be available on the technical report

Security requirements

- **NFR-13:** Client users will only have access to part of the system
- **NFR-14:** Admin user will have access to all the system
- **NFR-15:** Passwords will be encrypted by WPA.
- **NFR-16:** Application will store logs.

Maintainability and portability requirements

- **NFR-17:** Software will be modular to make it easier to expand into other universities and organizations.
- **NFR-18:** Application will save user data independently of the hardware used.

Resource requirements

- **NFR-19:** Application will be light-weight (less than 500M)

Performance requirements

- **NFR-20:** Response time will be fast (less than 1 second)

- **NFR-21:** Application will be designed for servers with enough capacity to handle multiple users accessing shared resources.

Availability requirements

- **NFR-22:** The application will need to be connected to the internet to work

Support requirements

- **NFR-23:** The application will follow standards to be easy to install and update.
- **NFR-24:** The application will be modular to help an easier maintainability.

Verification requirements

- **NFR-25:** The application will be transparent of errors to the user.

Legal requirements

- **NFR-26:** The application will be adapted to fulfill and follow all laws and restrictions of Spain, Madrid, and the UAM.

2.3 Deliverables

After this document is delivered, any modification or fix will be changed, and an updated revised version of the project plan will be delivered. This final project plan will not be changed during the development of the application.

As the development life cycle will be incremental, three different increments will be done with different functionalities implemented and goals reached. For each increment the following documents will be produced:

- Analysis document: Produced at the end of the analysis phase, documenting all the decisions taken for this phase.
- Design document: Produced at the end of the analysis phase, this document will document the design work done.
- User manual: Produced at the end of the coding and unit testing phase, this document will help the user understand how to use the application.
- Tech manual: Produced at the end of the coding and unit testing phase, this document will show all the technical documentation about the product.

For each iteration, the code of the corresponding version of the program will be delivered along with the executables to preview the product progress. At the final iteration, the material provided will be the final product.

3. Solution to Employ

ARCHITECTURE

The project will be designed using a multi-tier architecture:

- The data tier, in which we will manage the local database, access to the external Moodle UAM DB, and use of the Google Maps technology.
- The application tier, in which all the logic of the system will be stored: transactions, bookings, ratings, and use of the POS to exchange currency with the real world.
- The client tier, which will be composed by the GUI, different user forms, and FAQ.

PORTABILITY

The product will be designed for mobile devices, mainly Android and IOS powered, although a browser interface will also be available.

LIFE CYCLE AND DEVELOPMENT

The development team chose an incremental and iterative life cycle for the product. This approach will enable us to provide faster results, than can be later verified with the client. The client will be able to obtain functional increments of the final product faster and provide feedback, which is crucial since this project is pioneering on its field. There will be three increments: in the first one the users will be able to register and publish rides; the second one will enable them to book seats and perform payments between them; and the third one will cover the rating system and the incidents management.

Each increment will be composed by: Analysis, Design, Coding and Unit Testing, Integration Testing and Installation.

EXTERNAL SERVICES USED

- The application will access the verification page of Moodle UAM in order to validate the user credentials.
- For the transactions of real money and virtual credits, an online POS Gateway will be facilitated. We will be selecting PayPal for its convenience, card type support, and transaction speed.
- In order to obtain all the geolocalized data, Google Maps will be employed. We will use their APIs for displaying the map and routing users and car rides. For this both Google Maps SDK and their API DIRECTIONS will be used.

4. Project Management

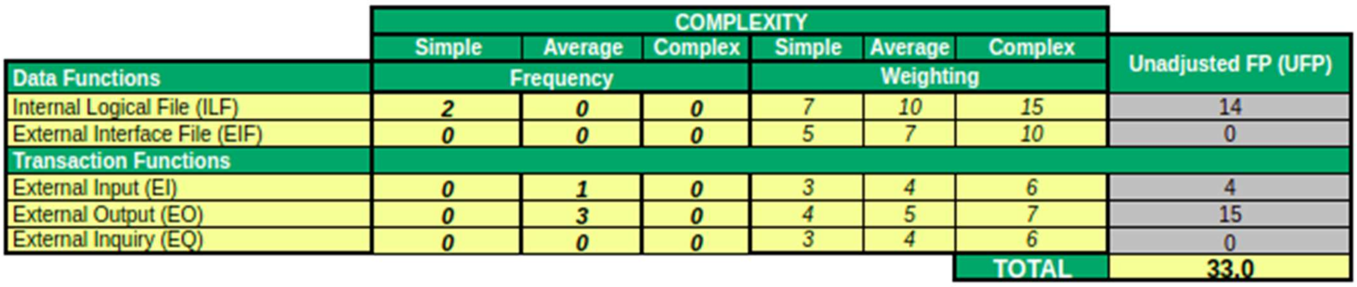
4.1 Estimations of the Software System

To make the application of estimation techniques of the project, the team has used the Function Points model. This model is based on the logical design to qualify the external functionality provided by the software. The model would provide metrics about productivity, quality, cost, documentation and code lines. In the implementation provided by the team the system is divided into several subsystems, for each of them the model was applied, considering the following parameters: external inputs (EI), external outputs (EO), external inquiry (EQ) and inner or external logical data (ILF/EIF). These parameters are used to obtain the Unadjusted Function Points (UFP), from which we can obtain the Adjustment Factor (AFP), by counting the number of DETs and FTRs. From all these it is obtained an estimation of the complexity of each subsystem.

4.1.1 Function Points

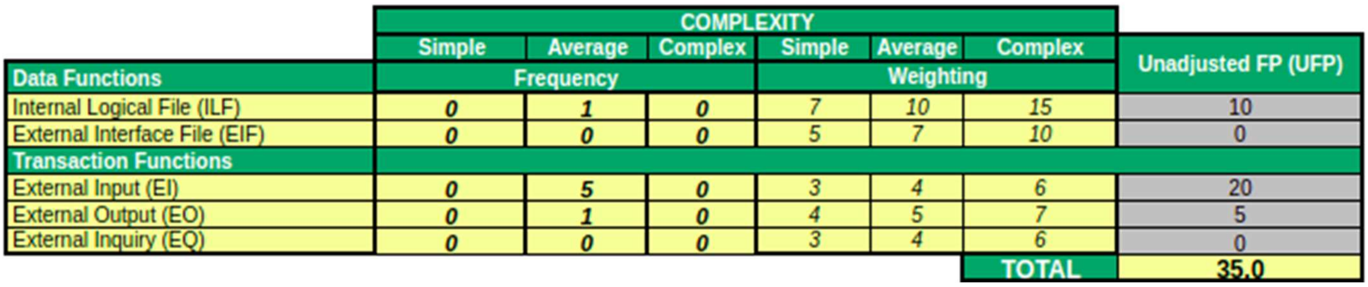
Following the obtained function points are presented.

4.1.1.1 Incident subsystem



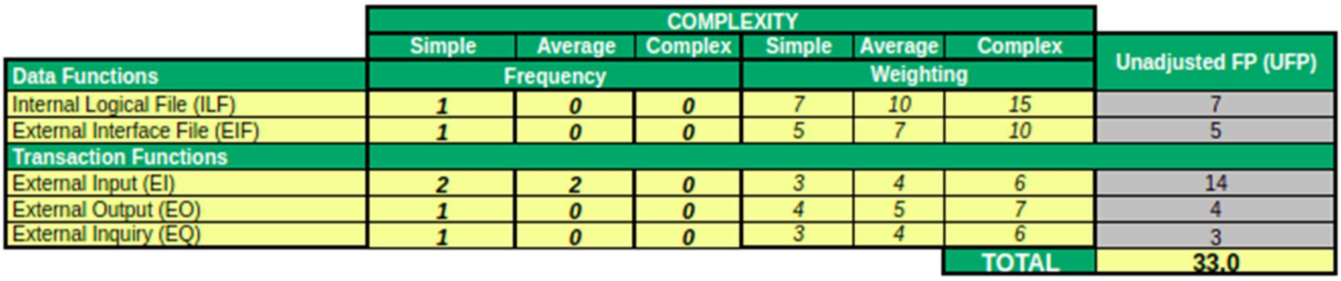
	COMPLEXITY						
	Simple	Average	Complex	Simple	Average	Complex	Unadjusted FP (UFP)
Data Functions	Frequency			Weighting			
Internal Logical File (ILF)	2	0	0	7	10	15	14
External Interface File (EIF)	0	0	0	5	7	10	0
Transaction Functions							
External Input (EI)	0	1	0	3	4	6	4
External Output (EO)	0	3	0	4	5	7	15
External Inquiry (EQ)	0	0	0	3	4	6	0
						TOTAL	33.0

4.1.1.2 Publication of Rides subsystem



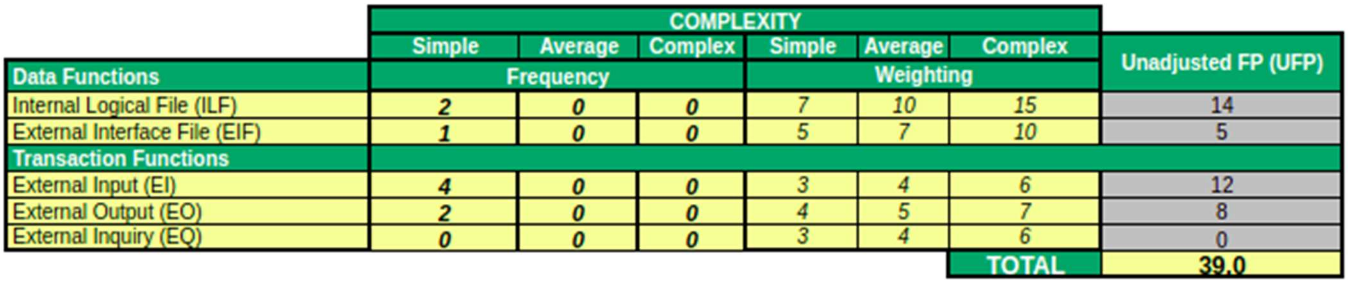
	COMPLEXITY						Unadjusted FP (UFP)
	Simple	Average	Complex	Simple	Average	Complex	
Data Functions	Frequency			Weighting			
Internal Logical File (ILF)	0	1	0	7	10	15	10
External Interface File (EIF)	0	0	0	5	7	10	0
Transaction Functions							
External Input (EI)	0	5	0	3	4	6	20
External Output (EO)	0	1	0	4	5	7	5
External Inquiry (EQ)	0	0	0	3	4	6	0
						TOTAL	35.0

4.1.1.3 User Management subsystem



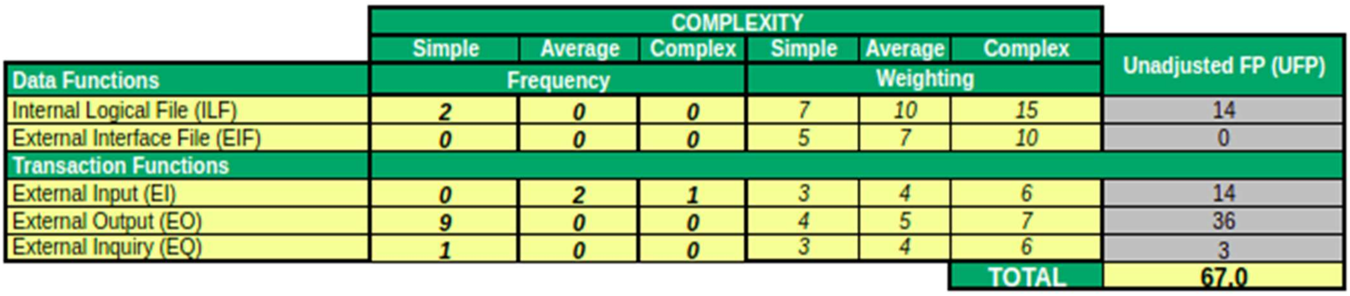
	COMPLEXITY						Unadjusted FP (UFP)
	Simple	Average	Complex	Simple	Average	Complex	
Data Functions	Frequency			Weighting			
Internal Logical File (ILF)	1	0	0	7	10	15	7
External Interface File (EIF)	1	0	0	5	7	10	5
Transaction Functions							
External Input (EI)	2	2	0	3	4	6	14
External Output (EO)	1	0	0	4	5	7	4
External Inquiry (EQ)	1	0	0	3	4	6	3
						TOTAL	33.0

4.1.1.4 Payment Management subsystem



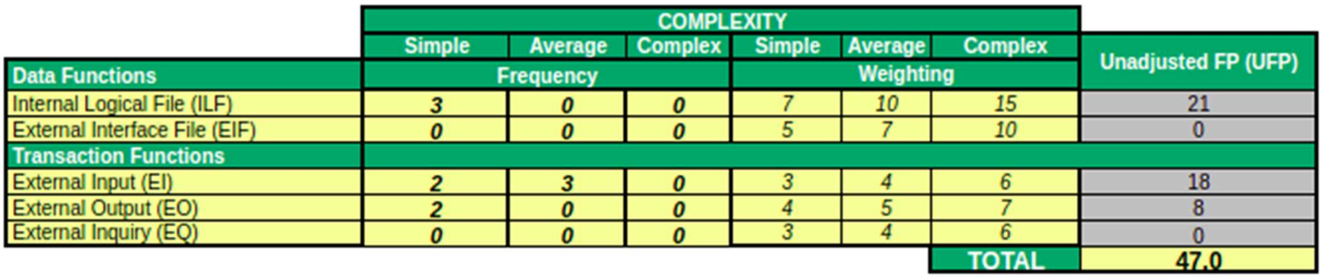
	COMPLEXITY						Unadjusted FP (UFP)
	Simple	Average	Complex	Simple	Average	Complex	
Data Functions	Frequency			Weighting			
Internal Logical File (ILF)	2	0	0	7	10	15	14
External Interface File (EIF)	1	0	0	5	7	10	5
Transaction Functions							
External Input (EI)	4	0	0	3	4	6	12
External Output (EO)	2	0	0	4	5	7	8
External Inquiry (EQ)	0	0	0	3	4	6	0
						TOTAL	39.0

4.1.1.5 Ride Seat Booking Management subsystem



	COMPLEXITY						Unadjusted FP (UFP)
	Simple	Average	Complex	Simple	Average	Complex	
Data Functions	Frequency			Weighting			
Internal Logical File (ILF)	2	0	0	7	10	15	14
External Interface File (EIF)	0	0	0	5	7	10	0
Transaction Functions							
External Input (EI)	0	2	1	3	4	6	14
External Output (EO)	9	0	0	4	5	7	36
External Inquiry (EQ)	1	0	0	3	4	6	3
						TOTAL	67.0

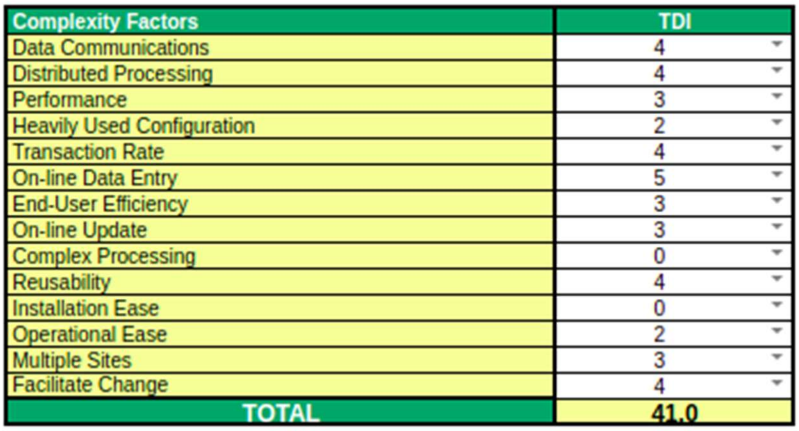
4.1.1.6 User Assessment subsystem



	COMPLEXITY						Unadjusted FP (UFP)
	Simple	Average	Complex	Simple	Average	Complex	
Data Functions	Frequency			Weighting			
Internal Logical File (ILF)	3	0	0	7	10	15	21
External Interface File (EIF)	0	0	0	5	7	10	0
Transaction Functions							
External Input (EI)	2	3	0	3	4	6	18
External Output (EO)	2	0	0	4	5	7	8
External Inquiry (EQ)	0	0	0	3	4	6	0
						TOTAL	47.0

4.1.2 Complexity factors

The following complexity factors were obtained for the whole system:

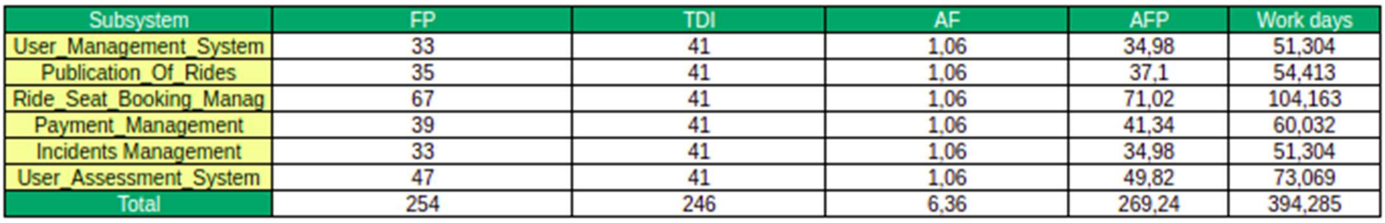


Complexity Factors	TDI
Data Communications	4
Distributed Processing	4
Performance	3
Heavily Used Configuration	2
Transaction Rate	4
On-line Data Entry	5
End-User Efficiency	3
On-line Update	3
Complex Processing	0
Reusability	4
Installation Ease	0
Operational Ease	2
Multiple Sites	3
Facilitate Change	4
TOTAL	41.0

See Appendix B to see details of each of them.

4.1.3 Grouped Summary

The following table contains a grouped summary of the results and a total sum of all the subsystems.



Subsystem	FP	TDI	AF	AFP	Work days
User_Management_System	33	41	1,06	34,98	51,304
Publication_Of_Rides	35	41	1,06	37,1	54,413
Ride_Seat_Booking_Manag	67	41	1,06	71,02	104,163
Payment_Management	39	41	1,06	41,34	60,032
Incidents Management	33	41	1,06	34,98	51,304
User_Assessment_System	47	41	1,06	49,82	73,069
Total	254	246	6,36	269,24	394,285

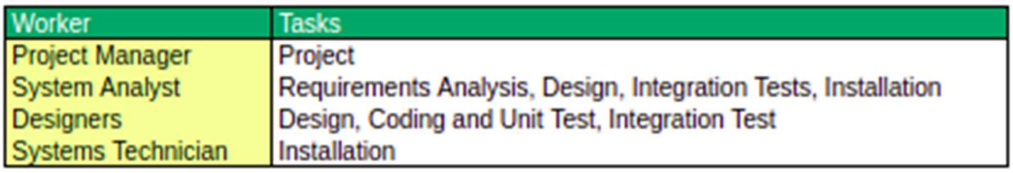
4.2 Organizational Structure

For this project the team would be:

- 1 system analyst who would be in charge of analyzing the requirements, design, integration tests and installation, in every iteration.
- 1 “senior” designer who would be in charge of designing the system design, coding and unit testing and integration tests.
- 2 “junior designers” who would be in charge of helping the “senior designer”.
- 1 systems technician who would be in charge of the installation of the system.
- 1 project manager who would be in charge of carrying out all the project management tasks.

4.3 Assigned Qualified Personnel

The team assignment to each task can be observe in the following table:

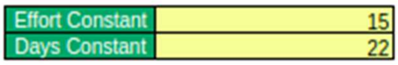


Worker	Tasks
Project Manager	Project
System Analyst	Requirements Analysis, Design, Integration Tests, Installation
Designers	Design, Coding and Unit Test, Integration Test
Systems Technician	Installation

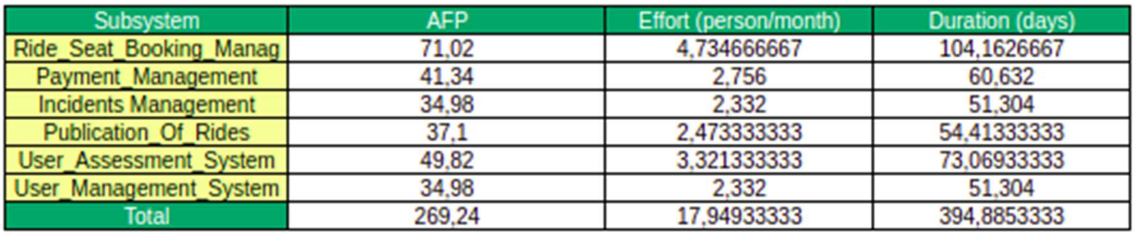
Go to appendix C to see a more detailed task distribution.

4.4 Time Management

The project is divided into 6 subsystems, which will be developed through 3 increments. Considering the working calendar from Monday to Friday, with 40 business hours and 22 working days/person-month. From historical data there is an estimate of 15 FP / person-month. With these data the following table is obtained:



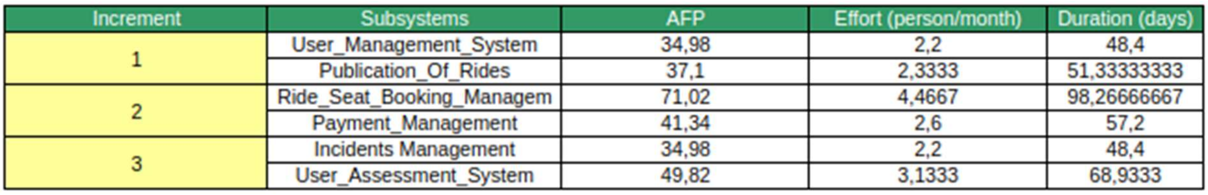
Effort Constant	15
Days Constant	22



Subsystem	AFP	Effort (person/month)	Duration (days)
Ride_Seat_Booking_Manag	71,02	4,734666667	104,1626667
Payment_Management	41,34	2,756	60,632
Incidents Management	34,98	2,332	51,304
Publication_Of_Rides	37,1	2,473333333	54,41333333
User_Assessment_System	49,82	3,321333333	73,06933333
User_Management_System	34,98	2,332	51,304
Total	269,24	17,94933333	394,8853333

The project's work would be distributed in 3 increments, in each one some subsystems would be developed, following a reasonable workflow. The effort is also taken into account when choosing each subsystem.

The following table is obtained:



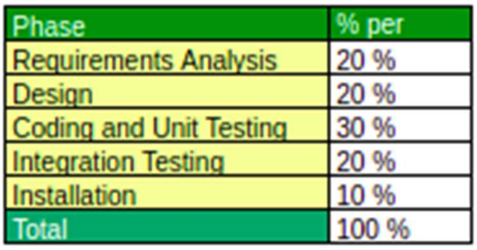
Increment	Subsystems	AFP	Effort (person/month)	Duration (days)
1	User_Management_System	34,98	2,2	48,4
	Publication_Of_Rides	37,1	2,3333	51,33333333
2	Ride_Seat_Booking_Managem	71,02	4,4667	98,26666667
	Payment_Management	41,34	2,6	57,2
3	Incidents Management	34,98	2,2	48,4
	User_Assessment_System	49,82	3,1333	68,9333

Iteration 1: The most basic subsystems are chosen, in this case, user management and publication of rides. Most of the other subsystems will interact with the chosen ones.

Iteration 2: The subsystems chosen are the ride seat booking management and payment management. The first one is the subsystems with the biggest effort, and also, is the second main functionality of the application (with the publication of rides). The second is joined with the first one, as it does not make sense to book and not interact with the credits needed to book.

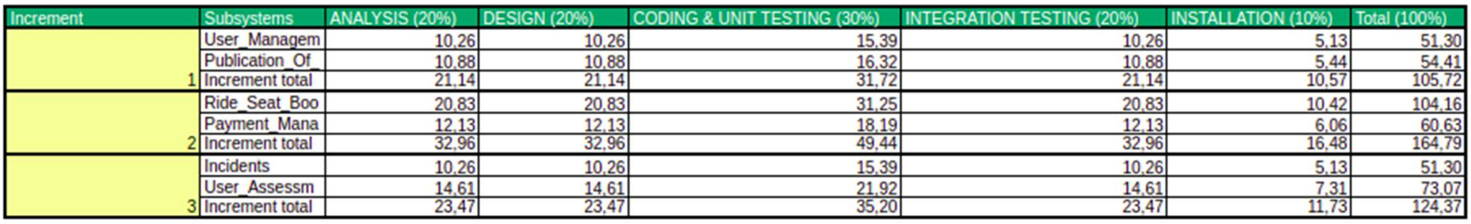
Iteration 3: In this final iteration, the incident management and user assessment subsystems would be implemented. This functionality is the least relevant among the rest.

The effort distribution for each increment, according to the historical database is the following:



Phase	% per
Requirements Analysis	20 %
Design	20 %
Coding and Unit Testing	30 %
Integration Testing	20 %
Installation	10 %
Total	100 %

With all this data we can estimate the task effort distribution for each subsystem and increment in the following table:



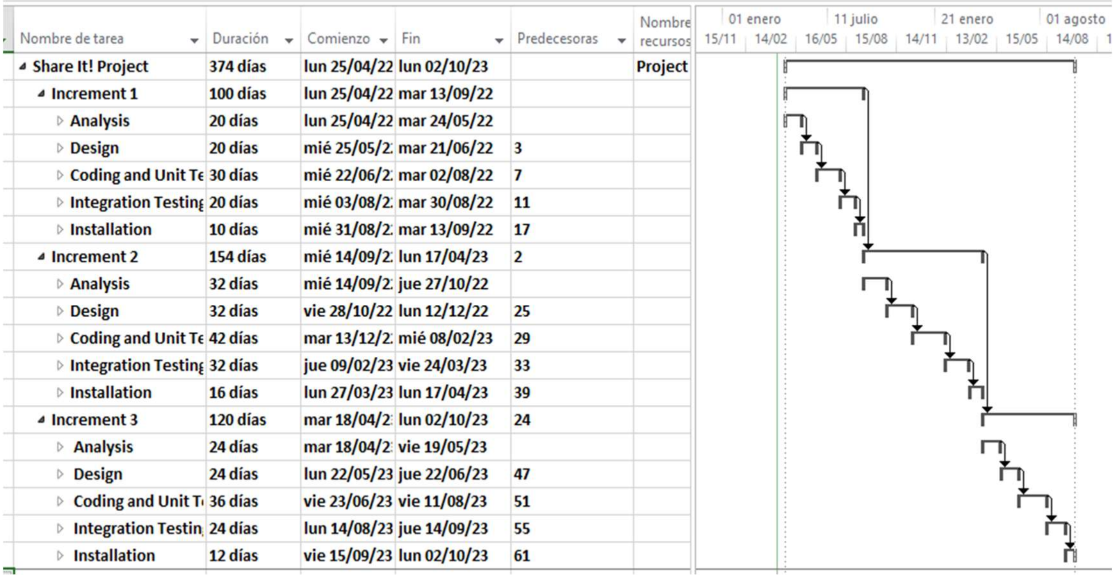
Increment	Subsystems	ANALYSIS (20%)	DESIGN (20%)	CODING & UNIT TESTING (30%)	INTEGRATION TESTING (20%)	INSTALLATION (10%)	Total (100%)
1	User_Managem	10,26	10,26	15,39	10,26	5,13	51,30
	Publication_Of_	10,88	10,88	16,32	10,88	5,44	54,41
	Increment total	21,14	21,14	31,72	21,14	10,57	105,72
2	Ride_Seat_Boo	20,83	20,83	31,25	20,83	10,42	104,16
	Payment_Mana	12,13	12,13	18,19	12,13	6,06	60,63
	Increment total	32,96	32,96	49,44	32,96	16,48	164,79
3	Incidents	10,26	10,26	15,39	10,26	5,13	51,30
	User_Assessm	14,61	14,61	21,92	14,61	7,31	73,07
	Increment total	23,47	23,47	35,20	23,47	11,73	124,37

A waterfall lifecycle is followed by the phases within the increment. The requirements analysis for the entire system/increment must be done in a unified manner. For each of the various subsystems examined at each increment, the design, coding, and unit testing processes can be completed in parallel. They'll get started after the requirements analysis is updated. After the subsystem design has been amended, each subsystem's implementation will begin; unit testing will follow once the implementation is complete. When all of the unit tests of the various subsystems have been performed for each increment, the integration testing phase will begin. After the integration tests have been changed, the installation process will begin.

Each intermediate and final product will have four milestones: the end of the requirements analysis, the end of the design, the end of integration tests, and the end of the installation.

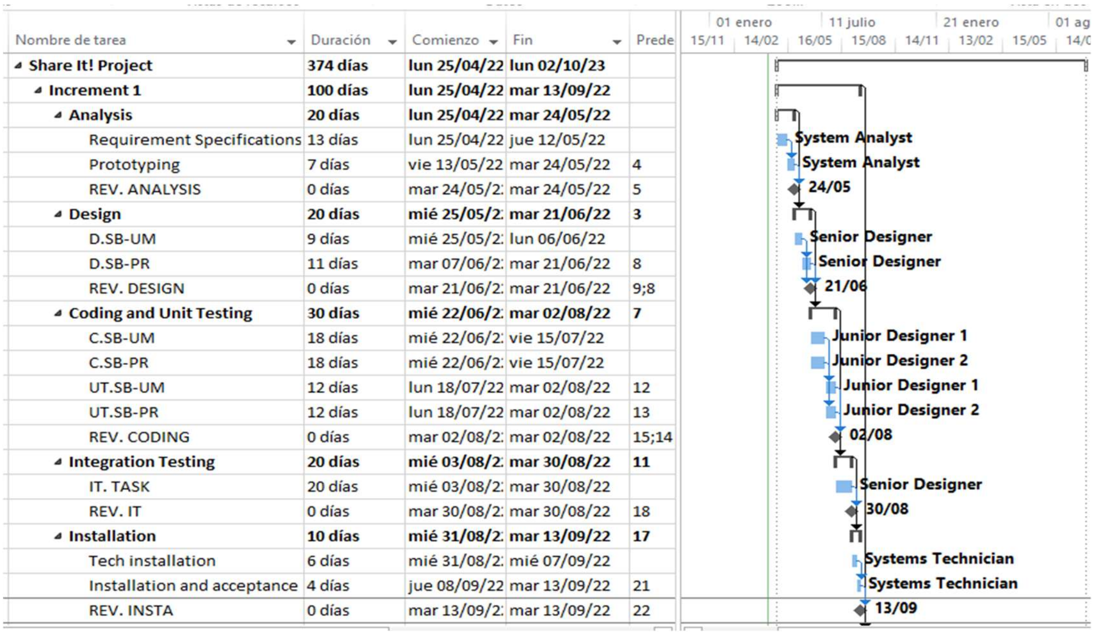
The following schedule is the product of a meticulous work distribution that reduces the amount of time required to complete the project. The resulting Gantt diagrams show the distribution.

Project Overview:



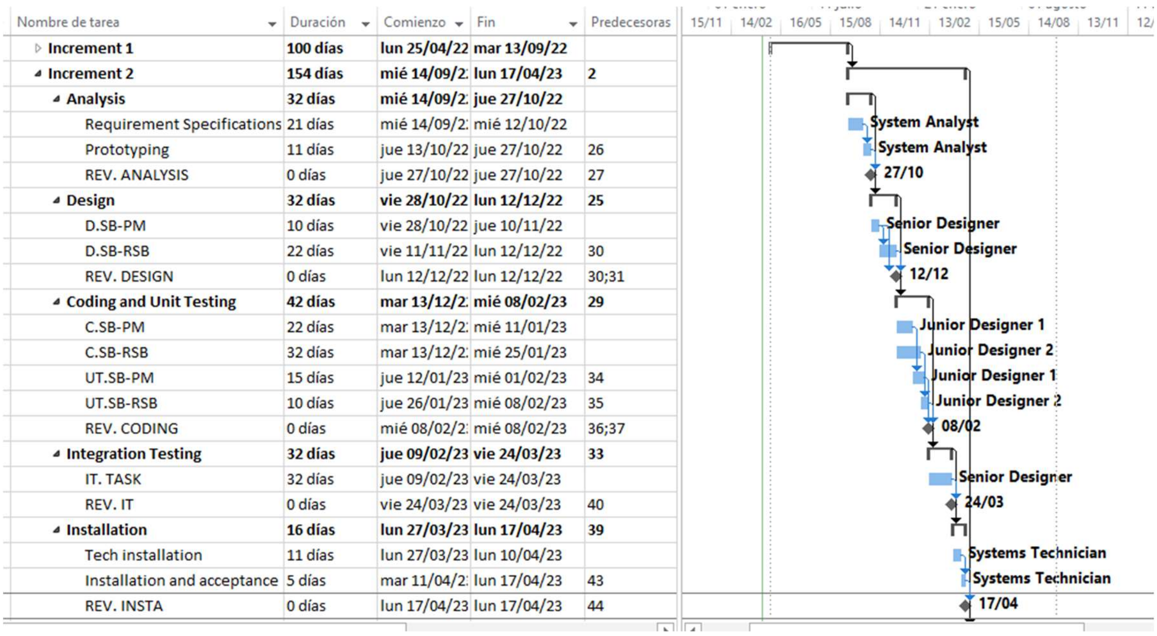
Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombre recursos
Share It! Project	374 días	lun 25/04/22	lun 02/10/23		Project
Increment 1	100 días	lun 25/04/22	mar 13/09/22		
Analysis	20 días	lun 25/04/22	mar 24/05/22		
Design	20 días	mié 25/05/2:	mar 21/06/22	3	
Coding and Unit Te	30 días	mié 22/06/2:	mar 02/08/22	7	
Integration Testing	20 días	mié 03/08/2:	mar 30/08/22	11	
Installation	10 días	mié 31/08/2:	mar 13/09/22	17	
Increment 2	154 días	mié 14/09/2:	lun 17/04/23	2	
Analysis	32 días	mié 14/09/2:	jue 27/10/22		
Design	32 días	vie 28/10/22	lun 12/12/22	25	
Coding and Unit Te	42 días	mar 13/12/2:	mié 08/02/23	29	
Integration Testing	32 días	jue 09/02/23	vie 24/03/23	33	
Installation	16 días	lun 27/03/23	lun 17/04/23	39	
Increment 3	120 días	mar 18/04/2:	lun 02/10/23	24	
Analysis	24 días	mar 18/04/2:	vie 19/05/23		
Design	24 días	lun 22/05/23	jue 22/06/23	47	
Coding and Unit Te	36 días	vie 23/06/23	vie 11/08/23	51	
Integration Testin	24 días	lun 14/08/23	jue 14/09/23	55	
Installation	12 días	vie 15/09/23	lun 02/10/23	61	

Increment 1:



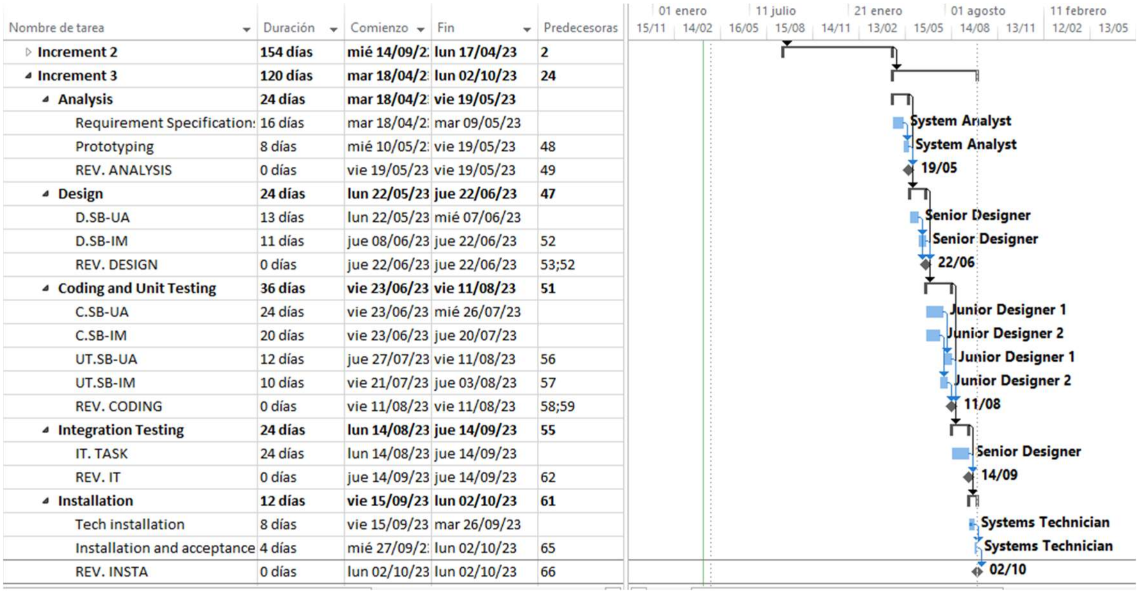
Nombre de tarea	Duración	Comienzo	Fin	Prede
Share It! Project	374 días	lun 25/04/22	lun 02/10/23	
Increment 1	100 días	lun 25/04/22	mar 13/09/22	
Analysis	20 días	lun 25/04/22	mar 24/05/22	
Requirement Specifications	13 días	lun 25/04/22	jue 12/05/22	
Prototyping	7 días	vie 13/05/22	mar 24/05/22	4
REV. ANALYSIS	0 días	mar 24/05/2:	mar 24/05/22	5
Design	20 días	mié 25/05/2:	mar 21/06/22	3
D.SB-UM	9 días	mié 25/05/2:	lun 06/06/22	
D.SB-PR	11 días	mar 07/06/2:	mar 21/06/22	8
REV. DESIGN	0 días	mar 21/06/2:	mar 21/06/22	9;8
Coding and Unit Testing	30 días	mié 22/06/2:	mar 02/08/22	7
C.SB-UM	18 días	mié 22/06/2:	vie 15/07/22	
C.SB-PR	18 días	mié 22/06/2:	vie 15/07/22	
UT.SB-UM	12 días	lun 18/07/22	mar 02/08/22	12
UT.SB-PR	12 días	lun 18/07/22	mar 02/08/22	13
REV. CODING	0 días	mar 02/08/2:	mar 02/08/22	15;14
Integration Testing	20 días	mié 03/08/2:	mar 30/08/22	11
IT. TASK	20 días	mié 03/08/2:	mar 30/08/22	
REV. IT	0 días	mar 30/08/2:	mar 30/08/22	18
Installation	10 días	mié 31/08/2:	mar 13/09/22	17
Tech installation	6 días	mié 31/08/2:	mié 07/09/22	
Installation and acceptance	4 días	jue 08/09/22	mar 13/09/22	21
REV. INSTA	0 días	mar 13/09/2:	mar 13/09/22	22

Increment 2:



Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
▷ **Increment 1**	**100 días**	**lun 25/04/22**	**mar 13/09/22**	
◢ **Increment 2**	**154 días**	**mié 14/09/2:**	**lun 17/04/23**	**2**
◢ **Analysis**	**32 días**	**mié 14/09/2:**	**jue 27/10/22**	
Requirement Specifications	21 días	mié 14/09/2:	mié 12/10/22	
Prototyping	11 días	jue 13/10/22	jue 27/10/22	26
REV. ANALYSIS	0 días	jue 27/10/22	jue 27/10/22	27
◢ **Design**	**32 días**	**vie 28/10/22**	**lun 12/12/22**	**25**
D.SB-PM	10 días	vie 28/10/22	jue 10/11/22	
D.SB-RSB	22 días	vie 11/11/22	lun 12/12/22	30
REV. DESIGN	0 días	lun 12/12/22	lun 12/12/22	30;31
◢ **Coding and Unit Testing**	**42 días**	**mar 13/12/2:**	**mié 08/02/23**	**29**
C.SB-PM	22 días	mar 13/12/2:	mié 11/01/23	
C.SB-RSB	32 días	mar 13/12/2:	mié 25/01/23	
UT.SB-PM	15 días	jue 12/01/23	mié 01/02/23	34
UT.SB-RSB	10 días	jue 26/01/23	mié 08/02/23	35
REV. CODING	0 días	mié 08/02/2:	mié 08/02/23	36;37
◢ **Integration Testing**	**32 días**	**jue 09/02/23**	**vie 24/03/23**	**33**
IT. TASK	32 días	jue 09/02/23	vie 24/03/23	
REV. IT	0 días	vie 24/03/23	vie 24/03/23	40
◢ **Installation**	**16 días**	**lun 27/03/23**	**lun 17/04/23**	**39**
Tech installation	11 días	lun 27/03/23	lun 10/04/23	
Installation and acceptance	5 días	mar 11/04/2:	lun 17/04/23	43
REV. INSTA	0 días	lun 17/04/23	lun 17/04/23	44

Increment 3:



Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
▷ **Increment 2**	**154 días**	**mié 14/09/2:**	**lun 17/04/23**	**2**
◢ **Increment 3**	**120 días**	**mar 18/04/2:**	**lun 02/10/23**	**24**
◢ **Analysis**	**24 días**	**mar 18/04/2:**	**vie 19/05/23**	
Requirement Specification:	16 días	mar 18/04/2:	mar 09/05/23	
Prototyping	8 días	mié 10/05/2:	vie 19/05/23	48
REV. ANALYSIS	0 días	vie 19/05/23	vie 19/05/23	49
◢ **Design**	**24 días**	**lun 22/05/23**	**jue 22/06/23**	**47**
D.SB-UA	13 días	lun 22/05/23	mié 07/06/23	
D.SB-IM	11 días	jue 08/06/23	jue 22/06/23	52
REV. DESIGN	0 días	jue 22/06/23	jue 22/06/23	53;52
◢ **Coding and Unit Testing**	**36 días**	**vie 23/06/23**	**vie 11/08/23**	**51**
C.SB-UA	24 días	vie 23/06/23	mié 26/07/23	
C.SB-IM	20 días	vie 23/06/23	jue 20/07/23	
UT.SB-UA	12 días	jue 27/07/23	vie 11/08/23	56
UT.SB-IM	10 días	vie 21/07/23	jue 03/08/23	57
REV. CODING	0 días	vie 11/08/23	vie 11/08/23	58;59
◢ **Integration Testing**	**24 días**	**lun 14/08/23**	**jue 14/09/23**	**55**
IT. TASK	24 días	lun 14/08/23	jue 14/09/23	
REV. IT	0 días	jue 14/09/23	jue 14/09/23	62
◢ **Installation**	**12 días**	**vie 15/09/23**	**lun 02/10/23**	**61**
Tech installation	8 días	vie 15/09/23	mar 26/09/23	
Installation and acceptance	4 días	mié 27/09/2:	lun 02/10/23	65
REV. INSTA	0 días	lun 02/10/23	lun 02/10/23	66

According to the planning, the expected length is 374 days.

4.5 Costs Management

The project will take into consideration the potential costs involved in its completion, whether via the employment of competent employees or the usage of development material.

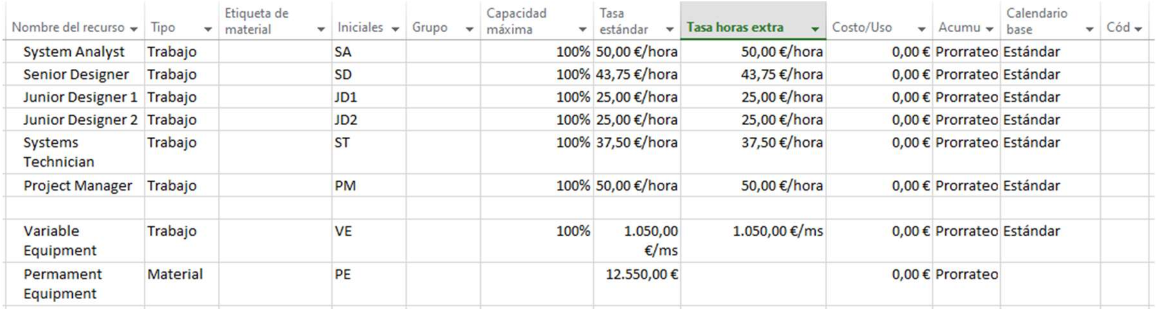
The following competent employees will have the assigned costs:

- System analyst: 400 euros/day.
- Senior designer: 350 euros/day.
- Junior designer: 200 euros/day.
- Systems technician: 300 euros/day.

Also, the development material, such as hardware resources and development software would have the following costs:

- Development equipment: 1050 euros/month.
- 3 workstations: 1650 euros * 3.
- Performance workstation: 3200 euros.
- Development environment: 1100 euros * 3.

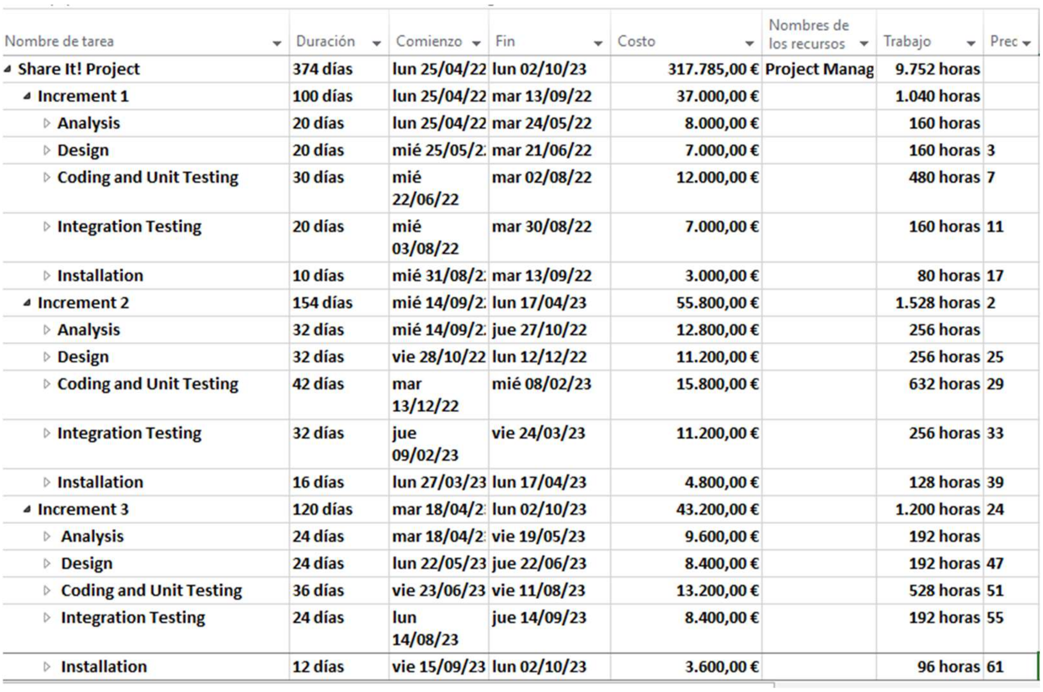
All the resources can be observed in the following table:



Nombre del recurso	Tipo	Etiqueta de material	Iniciales	Grupo	Capacidad máxima	Tasa estándar	Tasa horas extra	Costo/Uso	Acumu	Calendario base	Cód
System Analyst	Trabajo		SA		100%	50,00 €/hora	50,00 €/hora	0,00 €	Prorrateo	Estándar	
Senior Designer	Trabajo		SD		100%	43,75 €/hora	43,75 €/hora	0,00 €	Prorrateo	Estándar	
Junior Designer 1	Trabajo		JD1		100%	25,00 €/hora	25,00 €/hora	0,00 €	Prorrateo	Estándar	
Junior Designer 2	Trabajo		JD2		100%	25,00 €/hora	25,00 €/hora	0,00 €	Prorrateo	Estándar	
Systems Technician	Trabajo		ST		100%	37,50 €/hora	37,50 €/hora	0,00 €	Prorrateo	Estándar	
Project Manager	Trabajo		PM		100%	50,00 €/hora	50,00 €/hora	0,00 €	Prorrateo	Estándar	
Variable Equipment	Trabajo		VE		100%	1.050,00 €/ms	1.050,00 €/ms	0,00 €	Prorrateo	Estándar	
Permament Equipment	Material		PE			12.550,00 €		0,00 €	Prorrateo		

The permanent equipment would be a union of the 3 workstations, performance workstation and development environment.

The following table resumes the cost in each phase of each increment:



Nombre de tarea	Duración	Comienzo	Fin	Costo	Nombres de los recursos	Trabajo	Prec
◢ Share It! Project	374 días	lun 25/04/22	lun 02/10/23	317.785,00 €	Project Manag	9.752 horas	
◢ Increment 1	100 días	lun 25/04/22	mar 13/09/22	37.000,00 €		1.040 horas	
▷ Analysis	20 días	lun 25/04/22	mar 24/05/22	8.000,00 €		160 horas	
▷ Design	20 días	mié 25/05/2:	mar 21/06/22	7.000,00 €		160 horas	3
▷ Coding and Unit Testing	30 días	mié 22/06/22	mar 02/08/22	12.000,00 €		480 horas	7
▷ Integration Testing	20 días	mié 03/08/22	mar 30/08/22	7.000,00 €		160 horas	11
▷ Installation	10 días	mié 31/08/2:	mar 13/09/22	3.000,00 €		80 horas	17
◢ Increment 2	154 días	mié 14/09/2:	lun 17/04/23	55.800,00 €		1.528 horas	2
▷ Analysis	32 días	mié 14/09/2:	jue 27/10/22	12.800,00 €		256 horas	
▷ Design	32 días	vie 28/10/22	lun 12/12/22	11.200,00 €		256 horas	25
▷ Coding and Unit Testing	42 días	mar 13/12/22	mié 08/02/23	15.800,00 €		632 horas	29
▷ Integration Testing	32 días	jue 09/02/23	vie 24/03/23	11.200,00 €		256 horas	33
▷ Installation	16 días	lun 27/03/23	lun 17/04/23	4.800,00 €		128 horas	39
◢ Increment 3	120 días	mar 18/04/2:	lun 02/10/23	43.200,00 €		1.200 horas	24
▷ Analysis	24 días	mar 18/04/2:	vie 19/05/23	9.600,00 €		192 horas	
▷ Design	24 días	lun 22/05/23	jue 22/06/23	8.400,00 €		192 horas	47
▷ Coding and Unit Testing	36 días	vie 23/06/23	vie 11/08/23	13.200,00 €		528 horas	51
▷ Integration Testing	24 días	lun 14/08/23	jue 14/09/23	8.400,00 €		192 horas	55
▷ Installation	12 días	vie 15/09/23	lun 02/10/23	3.600,00 €		96 horas	61

The project's total cost would be 317.735,00 €.

4.6 Quality Management

See in the Quality Management Document [1].

4.7 Risks Management

See in the Risks Management Document [2].

4.8 Acquisitions Management

The company has all of the required tools to complete the job. The monthly costs and use are 1,050 euros, which includes both hardware and software.

Three workstations at a cost of 1,650 euros each are required to develop the project. For performance testing, an additional workstation of 3,200 euros is required.

Finally, at a cost of 1,100 euros per workstation, a new integrated development environment will also be purchased. This environment contains all of the software required for the project's lifespan.

4.9 Documentation Management

See in the Documentation Management Document [3].

5. Monitoring and Control

5.1 Configuration Management

The documents that can trigger the change control protocol are the following:

- Software requirements document
- Design documentation
- Source and executable code
- User and technical manual

If the need for a change in any of these documents is discovered by an employee, if he is not part of the development team, he will have to notify a development team member about such change, and when the need for change arrives to a development team member, he must fill out a report indicating the following:

- Change requested
- Problems that this needed change is generating (reason for the change)
- Documents that the change could affect
- Pros and cons of implementing the change

This report must be sent to the project manager who will evaluate the change and make a decision on whether or not it is worth it to make the change, in the process of that decision, the project manager will get the most fit employer for managing such type of change and name him the change supervisor. The change supervisor will make a more detailed report reviewing the proposed change and make an estimation on the effort of completing this task. If the change affects the development of the product or the final product itself, a meeting with the client will be made to make them understand what that change means, the cost and effort that it would suppose and the pros and cons of such change. Once and if the final decision of accepting the change is made, the change supervisor along with the team manager will be in charge of making a team with the necessary personnel to make the change and they will start working on it. The change supervisor will be in charge of managing the team and checking all the documents are uploaded correctly and once the change has been made, he will generate a report with all the details of the change and the effort and working days spent on it. Then a meeting will take place analyzing where the error came from and making a review on the change implementation (checking whether the effort estimation was accurate, if there were any complications, how could that process be improved…) and the report will be sent to the project manager.

5.2 Progress Monitorization

One first meeting with the client will be made to present the project and sign the contract for its development, after this, the increments start. There will be a total of three increments and after each, a meeting with the client will be made to present the state of the product and present a report on the increment explaining if there were corrections and overall state of the project. This way after the third and last increment the client will have a fully developed project and a project.

In each Increment there will be two kinds of weekly meetings one between the subproject managers and their team where the individual progress on the project will be measured, the team will explain the problems they had and future intentions for the next week, in this meeting the subproject progress will be calculated, and a weekly report will be made with all this data. The other weekly meeting will be between the subproject managers and the project manager where the progress of each subproject will be shown and compared to the expected one. There the project manager, helped by the opinion of the according subproject manager, will make the decision on whether the project planning needs a correction or not depending on the expected and real progress.

5.3 Verifications at each Phase

There will be a meeting after the end date of each phase where the product developed until then must be delivered to the client, however before this meeting, a review on the project will be made where the following aspects will be checked:

- The requirements will be validated checking the following: if they are correct, if they have been completed, if they are ambiguous and if they are traceable
- The design document will be reviewed trying to accomplish the maximum cohesion and minimum coupling possible. In each module the extensibility, reusability and flexibility will be reviewed.
- For the source and executable code, they will be evaluated using unit and integration tests.
- Finally, the user and technical manual will be read both by the team developers and people external from it so as to get an inside and outside perspective and be able to identify problems such as inconsistencies, orthographic and grammatical correctness and missing information.

5.4 Testing and Validation

After the coding phase on each increment a series of unit tests will be created so as to identify every possible error in the analysis, design or codification stages.

Black box testing will be made throughout the whole of the project and for the most complicated and important parts (to be determined in each design phase), white box tests will also be made so as to ensure their perfect performance.

Once the unit testing is done, the integration testing will be made where the different modules developed will be tested in combination following a strategy determined in the design phase.

After the realization of all the previous tests, what follows is the validation test where the software as a whole will be delivered to the client so it can be checked for inconsistencies with the requirements initially proposed. After that, finally, an acceptance test will be made by the user where the software will be checked so as to know if it is ready to be a final product.

After having realized the black box, white box tests or integration tests, the according subproject manager will be responsible for generating a report explaining the tests made in his subproject until then along with the results obtained. The subproject manager will have to show the report in the increment review meeting with the project manager. For the final tests (validation and acceptance), the project manager will be present in the meeting with the client so as to explain any upcoming doubts, note possible changes requested, and clarify details of the project if necessary.

6. Conclusions

This document shows the plan to develop a project to create a software application to provide a service of car sharing for rides for teacher and students to and from the UAM. The project plan includes the estimation of costs, resources, effort and time of the project, as well as the associated planning of them.

A Gantt diagram was created to estimate the cost and time of development, and with our current team the expected time to provide the software is 374 days, with a total cost of 317.785,00€, if the project plan is followed.

The development will have three iterations, each of them ending with the part of the product and functionality implemented so far, and the complete documentation of the work will be provided.

The purpose of this project plan document is to provide a transparent, organized and clear display on how the project is considered to be done in a viable, fast and efficient way, to provide the expected results and accomplish the goals aimed for.

APPENDICES

Appendix A: Function Points.

In this appendix we will discuss where the different function points for each subsystem come from.

Users Management Subsystem

- **FR-UM-1:**
  EI-User Log In
  DET:
  - Username input box
  - Password input box
  - Submit button

  FTR:
  - User data
  - User profile

  Conclusion: The complexity is Low with 3 DET and 2 FTR

- **FR-UM-1:**
  EQ-System verification
  DET:
  - Username
  - Password

  FTR:
  - User data
  - User profile

  Conclusion: The complexity is Low with 2 DET and 2 FTR

- **FR-UM-2:**
  EI-User Logout
  DET:
  - Logout button
  - User session

  FTR:
  - User data
  - User profile

  Conclusion: The complexity is Low with 2 DET and 2 FTR

- **FR-UM-3:**
  EI-User Register
  DET:
  - Username input box
  - Password input box

  FTR:
  - User data
  - User profile
  - Balance data

  Conclusion: The complexity is Medium with 2 DET and 3 FTR

- **FR-UM-5:**
  EO-Look driver profile
  DET:
  - Drive information
  - Driver information
  - Submit button

  FTR:
  - Driver profile

  Conclusion: The complexity is Low with 3 DET and 1 FTR

- **FR-UM-6:**
  EI-Send Message
  DET:
  - User information
  - Receiver information
  - Message
  - Submit button

  FTR:
  - User profile
  - Receiver Profile
  - User Messages
  - Receiver Messages

  Conclusion: The complexity is Medium with 4 DET and 4 FTR

ILF

- **ILF-User Management**
  DET:
  - User Email
  - User Password
  - User Profile
  - User Messages

  RET:
  - Mandatory subgroup

  Conclusion: The complexity is Low with 4 DET and 1 RET

EIF

- **EIF-System verification**
  DET:
  - User Email
  - User Password

  RET:
  - Mandatory subgroup

  Conclusion: The complexity is Low with 2 DET and 1 RET

User Assessment Subsystem

- **FR-UA-1:**
  EI-Rate passengers being driver
  DET:
  - User reviewed information
  - Ride information
  - Rate input box
  - Confirmation button

  FTR:
  - Rider profile
  - Passenger profile
  - Passenger rating

Conclusion: The complexity is Medium with 4 DET and 3 FTR

- **FR-UA-2:**
  EI-Passenger qualities and observations
  DET:
  ○ Profile of the passenger selected
  ○ Dropdown with a set of qualities to choose from
  ○ Open text field
  ○ Submit button
  FTR:
  ○ Rider profile information
  ○ Passenger profile information
  ○ Ride information

  Conclusion: The complexity is Medium with 4 DET and 3 FTR

- **FR-UA-3:**
  EI-Review fellow passengers
  DET:
  ○ Profile of the passenger selected
  ○ Open text field
  ○ Submit button
  FTR:
  ○ User information
  ○ Other passengers profile information
  ○ Ride information

  Conclusion: The complexity is Medium with 3 DET and 3 FTR

- **FR-UA-4:**
  EO-Block poor users
  DET:
  ○ Message describing the reason for the block of the account.
  ○ Button to make a claim for the blocking.
  ○ A link to the apps FAQ section.
  FTR:
  ○ User profile information
  ○ Blocking information

  Conclusion: The complexity is Low with 3 DET and 2 FTR

- **FR-UA-5:**
  EI-File claim for blocked account
  DET:
    - Username
    - Description of the block
    - Open text field to explain the claim
    - Submit button

  FTR:
    - User profile information
    - Blocking information

  Conclusion: The complexity is Low with 4 DET and 2 FTR

- **FR-UA-6:**
  EO-Notification about the resolution of the blocked account claim
  DET:
    - Message with the verdict of the claim
    - Continue button

  FTR:
    - User account information
    - Blocking information

  Conclusion: The complexity is Low with 2 DET and 2 FTR

- **FR-UA-7:**
  EI-Admin can manage the claims
  DET:
    - Profiles of the claimants
    - List with all the claims

  FTR:
    - Profile information of the claimants
    - Blocking information

  Conclusion: The complexity is Low with 2 DET and 2 FTR

ILF

- **ILF-Reviews-And-Observations**
  DET:
  - Reviewer user
  - Reviewed user
  - Rating
  - Observation’s text
  - Qualities

  RET:
  - Mandatory subgroup
  - Optional subgroup: Reviewed user, Rating
  - Optional subgroup: Reviewed user, Observation’s text, Qualities

  Conclusion: The complexity is Low with 5 DET and 3 RET

- **ILF-Blocked-User**
  DET:
  - Blocked user

  RET:
  - Mandatory subgroup

  Conclusion: The complexity is Low with 1 DET and 1 RET

- **ILF-Claims**
  DET:
  - Reviewer user
  - Claim text
  - Status

  RET:
  - Mandatory subgroup

  Conclusion: The complexity is Low with 3 DET and 1 RET

Publication of Rides Subsystem

- **FR-PR-1:**
  EI-Publish ride
  DET:
    - User publishing information
    - Publish button
    - Price number input box
    - Seats number input box
    - Comments textbox
    - Slider tab from/to UAM
    - Button to the location page
    - Day, Month and Year number box
    - Recurring slider
    - Car type
    - Smoking filter
    - Weekday for recurrence buttons

  FTR:
    - Rider profile
    - User published rides
    - Rides details

  Conclusion: The complexity is Medium with 10 DET and 3 FTR

- **FR-PR-2:**
  EI-Modify ride seats
  DET:
    - User publishing information
    - Passengers' information
    - Save changes button
    - Cancel button
    - Seats number input box

  FTR:
    - Rider profile
    - User published rides
    - Ride details
    - Users' passengers profiles

  Conclusion: The complexity is Medium with 5 DET and 4 FTR

- **FR-PR-3:**

EI-Modify ride details

DET:

- User publishing information
- Passengers’ information
- Save changes button
- Cancel button
- Price number input box
- Seats number input box
- Comments textbox
- Slider tab from/to UAM
- Button to the location page
- Day, Month and Year number box
- Recurring slider
- Weekday for recurrence buttons
- Cancel edit button
- Keep editing button
- Error message
- Car type
- Smoking filter

FTR:

- Rider profile
- User published rides
- Ride details
- Users’ passengers profiles

Conclusion: The complexity is Medium with 15 DET and 4 FTR

- **FR-PR-4:**

  EI-Cancel ride

  DET:

  - User publishing information
  - Passengers’ information
  - Cancel button
  - Cancel ride button
  - Error message

  FTR:

  - Rider profile
  - User published rides
  - Ride details
  - Users’ passengers profiles

  Conclusion: The complexity is Medium with 5 DET and 4 FTR

- **FR-PR-5:**
  EO-See published rides
  DET:
    - User publishing information
    - Passengers' information
    - Cancel button
    - Cancel ride button
    - Error message

  FTR:
    - Rider profile
    - User published rides
    - Ride details
    - Users' passengers profiles

  Conclusion: The complexity is Medium with 5 DET and 4 FTR

- **FR-PR-5.1:**
  EI-Confirm ride made
  DET:
    - Confirm ride message
    - Ride code input text
    - Ride details

  FTR:
    - Rider profile
    - Ride details

  Conclusion: The complexity is Medium with 3 DET and 2 FTR

ILF

- **ILF-Publish ride**
  DET:
    - Publishing user
    - Ride date
    - Ride time
    - Ride price
    - Ride seats
    - Ride Comments
    - Ride destiny
    - Ride origin
    - Ride is recurring
    - Ride recurring days
    - Ride status

- Ride ID
- Ride car type
- Ride smoking filter
- Ride confirmation codes
- Ride passenger users

RET:

- Mandatory subgroup
- Optional subgroup: Publisher user, ride date, ride origin, ride destiny, ride price, ride time
- Optional subgroup: ride car type
- Optional subgroup: ride car type, Ride smoking filter
- Optional subgroup: Ride date, ride time
- Optional subgroup: ride price
- Optional subgroup: ride seats
- Optional subgroup: ride publishing user, ride passenger users, ride confirmation codes

Conclusion: The complexity is Medium with 16 DET and 8 RET

Ride Seat Booking Management Subsystem

- **FR-RSB-1:**
  EI-See-available-rides:
  DET:
  - Button to see available rides
  - Pick-up point
  - Date
  - Time
  - Arrival point
  - Box for the filters

  FTR:
  - Rides information
  - Users' information

  Conclusion: The complexity is Medium with 6 DET and 2 FTR

- **FR-RSB-1:**

EO-See-available-rides:
DET:

  - Available rides
  - Available rides according to filter

FTR:

  - Rides information
  - Users' information

Conclusion: The complexity is Low with 2 DET and 2 FTR

- **FR-RSB-2:**
  EO-See-ride-details:
  DET:
  - Button to see ride details
  - Id of ride
  - Description of the ride

  FTR:
  - Rides information

  Conclusion: The complexity is Low with 3 DET and 1 FTR

- **FR-RSB-3:**
  EI-Book-ride:
  DET:
  - Button to make booking
  - Pick-up point
  - Date
  - Time
  - Arrival point
  - Filters
  - Button of confirmation
  - Button to make recurring booking

  FTR:
  - Users' information
  - Rides information
  - Booking's information

  Conclusion: The complexity is High with 8 DET and 3 FTR

- **FR-RSB-3:**

EO-Book-ride:
DET:
- ○ Message indication booking made successfully or not

FTR:
- ○ Users' information
- ○ Rides information
- ○ Booking's information

Conclusion: The complexity is Low with 1 DET and 3 FTR

- **FR-RSB-3:**
  EO-Total-Cost:
  DET:
  - ○ Message indicating total cost

  FTR:
  - ○ Users' information
  - ○ Rides information
  - ○ Booking's information

  Conclusion: The complexity is Low with 1 DET and 3 FTR

- **FR-RSB-3:**
  EO-Verify-Credit:
  DET:
  - ○ Message indicating true or false

  FTR:
  - ○ Users' information

  Conclusion: The complexity is Low with 1 DET and 1 FTR

- **FR-RSB-3:**
  EO-Occupation:
  DET:
  - ○ Message indicating true or false

  FTR:
  - ○ Rides information

Conclusion: The complexity is Low with 1 DET and 1 FTR

- **FR-RSB-4:**

EI-Cancel-Booking:
DET:

  - Button to cancel booking.

FTR:

  - Users' information
  - Rides information
  - Booking's information

Conclusion: The complexity is Medium with 1 DET and 3 FTR

- **FR-RSB-4:**
  EO-Cancel-Booking:
  DET:
  - Message indicating cancelation of a booking made successfully or not

  FTR:
  - Users' information
  - Rides information
  - Booking's information

  Conclusion: The complexity is Low with 1 DET and 3 FTR

- **FR-RSB-3:** Passengers cannot reserve seats on different rides that overlap in time.
  EO-Reserve-Booking-NotValid:
  DET:
  - Text indicating error

  FTR:
  - Users' information
  - Booking's information

  Conclusion: The complexity is Low with 1 DET and 2 FTR

- **FR-RSB-6:**
  EO-See-rides-reserved:
  DET:
  - Rides reserved box

  FTR:
  - User information
  - Booking information

  Conclusion: The complexity is Low with 1 DET and 2 FTR

- **FR-RSB-7:**
  EO-Receive-notification-booking:
  DET:
  - Notification related to a booking

  FTR:
  - User information
  - Booking information

  Conclusion: The complexity is Low with 1 DET and 2 FTR

ILFs:

- **ILF-RSB:**
  DET:
  - Id of booking
  - Id of ride
  - Id of user
  - Pick-up point
  - Date
  - Time
  - Arrival point
  - Cost
  - VPA
  - Filters (Optional)

  RET:
  - Mandatory attributes
  - Optional attributes

  Conclusion: The complexity is Low with 10 DET and 3 RET

- **FR-RSB-6:**
  ILF-See-rides-reserved:
  DET:
  - Id of user
  - Id of rides

  RET:
  - Mandatory attributes

  Conclusion: The complexity is Low with 2 DET and 1 FTR

Payment Management Subsystem

- **FR-PM-1:**

  EI-Form to add a new credit card

  DET:
  - Credit card number
  - Card owner name
  - Expiration date
  - Card pin
  - Submit button

  FTR:
  - User information

  Conclusion: The complexity is Low with 5 DET and 1 FTR

- **FR-PM-1:**

  EI-Balance section request

  DET:
  - Button to balance section

  FTR:
  - User information

  Conclusion: The complexity is Low with 1 DET and 1 FTR

- **FR-PM-1:**

  EO-Balance section page display

  DET:
  - Available credits
  - Blocked credits
  - Total credits
  - Transfer credits button
  - Reimburse credits button

  FTR:

- VPA information
- User information

Conclusion: The complexity is Low with 5 DET and 2 FTR

- **FR-PM-1:**

  EO-Summary of credit cards for a user

  DET:

  - Card number
  - Card expiration date
  - Card type
  - Delete card button
  - Add card button

  FTR:
  - VPA information
  - User information

  Conclusion: The complexity is Low with 5 DET and 2 FTR

- **FR-PM-3:**

  EI-Form to transfer credits to the VPA

  DET:
  - Card to be used
  - Amount of credits
  - Submit button

  FTR:
  - VPA information
  - User information

  Conclusion: The complexity is Low with 3 DET and 2 FTR

- **FR-PM-4:**

EI-Form to reimburse credits from the VPA

DET:

- Card to be used
- Amount of credits
- Submit button

FTR:

- VPA information
- User information

Conclusion: The complexity is Low with 3 DET and 2 FTR

- **ILF:**

ILF-VPA

DET:

- Available credits
- Blocked credits
- Total credits
- User info
- Credit card

RET:

- Mandatory information all of the above
- Extra credit cards

Conclusion: The complexity is Low with 5 DET and 2 RET

- **ILF:**

ILF-User credit card

DET:

- Credit card number
- Card owner name
- Expiration date
- Card pin

RET:

- One subgroup, all are mandatory

Conclusion: The complexity is Low with 4 DET and 1 RET

- **EIF:**

  EIF-POS transaction terminal

  DET:
    - Payer banking information
    - Receiver banking information
    - Amount of credits
    - Product concept

  RET:
    - One subgroup, all are mandatory

  Conclusion: The complexity is Low with 4 DET and 1 RET

Incidents Management Subsystem

- **FR-IM-1:**
  EI-Incident Form
  DET:
    - Incident type dropdown
    - The description box
    - The user to be selected from the ride
    - Button to submit the form.

  FTR:
    - The ride information
    - The defendant user information
    - The claimant user information

  Conclusion: The complexity is Medium with 4 DET and 3 FTR

- **FR-IM-1:**
  EO-Notification of registered incident

DET:

- It has the text with the description again
- The incident id
- The names of the claimant
- and defendant
- The status as *issued*

FTR:

- Incident information
- Claimant user information
- Defendant user information
- Ride information

Conclusion: The complexity is Medium with 5 DET and 4 FTR

- **FR-IM-4:**
  EO-Notification about status update
  DET:
  - It will have the incident id
  - The old status
  - The new status
  - The names of the claimant
  - and defendant

  FTR:
  - Incident information
  - Claimant user information
  - Defendant user information
  - Ride information

  Conclusion: The complexity is Medium with 5 DET and 4 FTR

- **FR-IM-5:**
  EO-Notification about the resolution

DET

- ○ It will have the incident id
- ○ The resolution decision
- ○ The names of the claimant
- ○ and defendant
- ○ The number of Virtual credits resolved.

FTR

- ○ Incident information
- ○ Claimant user information
- ○ Defendant user information
- ○ Ride information

Conclusion: The complexity is Medium with 5 DET and 4 FTR

- **ILF:**
  ILF-IM
  DET:
  - ○ Incident id
  - ○ The name of the claimant
  - ○ Defendant name
  - ○ Category of the incident
  - ○ Status of the incident
  - ○ Description of the incident
  - ○ Claimant user info
  - ○ Defendant user info
  - ○ Ride info

  RET:
  - ○ One subgroup all are mandatory

  Conclusion: The complexity is Low with 9 DET and 1 RET

- **ILF:**
  ILF-IM-Resolution-Document
  DET:
  - ○ Incident id
  - ○ The name of the claimant
  - ○ Defendant name
  - ○ Category of the incident
  - ○ Description of the incident
  - ○ Resolution status

- Beneficiary user
- Amount of money granted
- Claimant user info
- Defendant user info
- Ride info

RET:

- Mandatory subgroup
- Optional subgroup: Beneficiary user, Amount of money granted

Conclusion: The complexity is Low with 11 DET and 2 RET

Appendix B: Complexity Factors.

In this appendix we will discuss where the different values for the complexity factors (or General System Features) come from. These values range from zero to five.

Data Communications: 4

The team thinks data communications is a 4 because in the statement it is said that the communication is online, the team assumes one protocol used in the communication. Also, there would be several users which means more than one front-end computer.

Distributed Processing: 4

The team thinks distributed processing is a 4 because as the statement says, for all cases data input, updates and queries shall be through online processes, so data transfer would be online and a distributed process as the application would have three-tier architecture.

Performance: 3

The team thinks performance is a 3 because as the statement says, the application expects concurrent users to be of medium-high size; therefore, special considerations shall be required regarding the performance. There is no information about the CPU in the statement so no special treatment shall be made.

Heavily used configuration: 2

According to the requirements, the application has some operative restrictions, but they are not of a very high exigence and can be fulfilled easily

Transaction rate: 4

According to the requirements special considerations shall be required concerning performance and security of transactions so the transaction rate must be analyzed, so the transaction rate that accomplishes this without overpassing the requirements is number 4.

On-line data entry: 5

Since it is an interactive application, most of the elements are going to be interactive such as buttons, text fields, sliders… So, the app will have a high percentage of interactive elements.

End User Efficiency: 3

The application uses most of the user efficiency functions (more than 6). There are not specific user efficiency requirements that make it necessary to design tasks that take into account human factors.

On-line Update: 3

The application updates logical files online constantly, to maintain all users synchronized with each other. Protection against data loss has not been especially designed in the system.

Complex Processing: 0

There is no need for complex processing at all in the system.

Reusability: 4

This factor takes a value of 4 since the code is meant to be reused. This project pioneers among its field, which is carsharing inside the university. It was thought that the code may be reused by other universities or institutions to implement their own car sharing platform.

Installation Ease: 0

The application will not require any special consideration or developments for the installation, as mentioned in the statement. The application will be a web service and the interface will be the browser.

Operational Ease: 2

The application will not depend on papers or tapes for recovery issues. The database will be backed up in auxiliary storage and recovery will be manual.

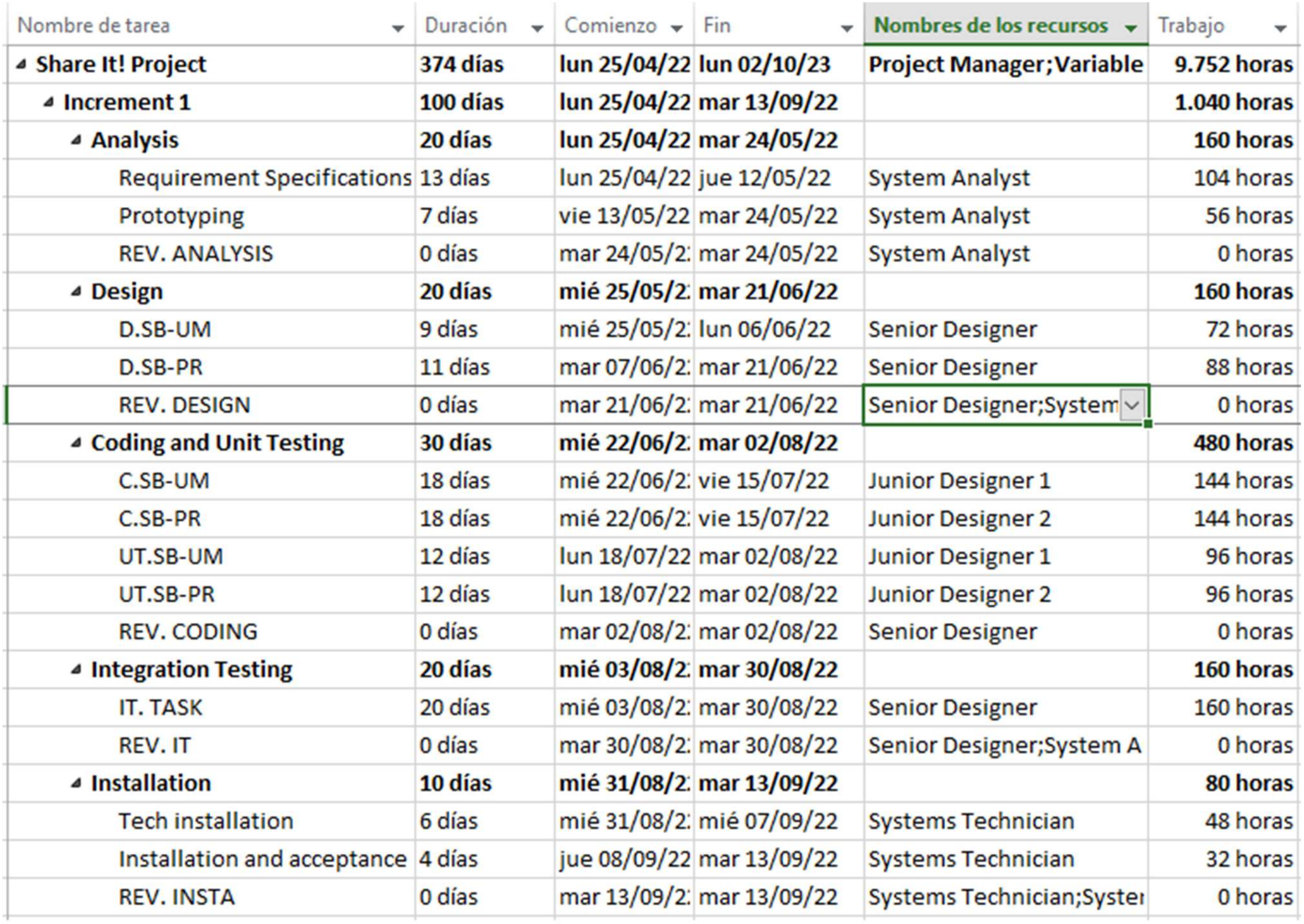
Multiple Sites: 3

The application is designed to work on different types of hardware (Computers, phones, etc.), but documentation and support plans are not provided.

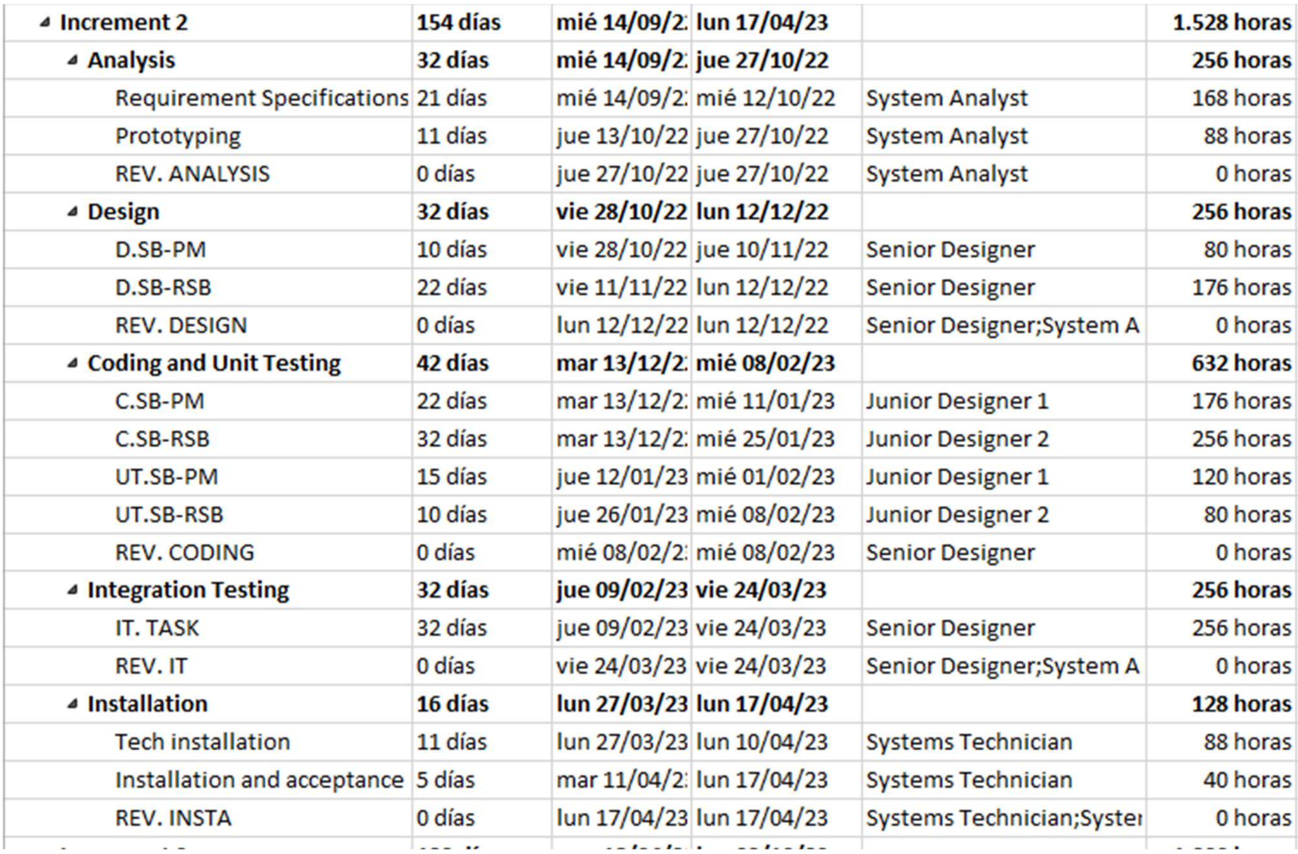
Facilitate change: 4

The complexity of the queries will not be too high, and the values must be updated instantly so as to avoid problems like concurrent reservations of the same spot.

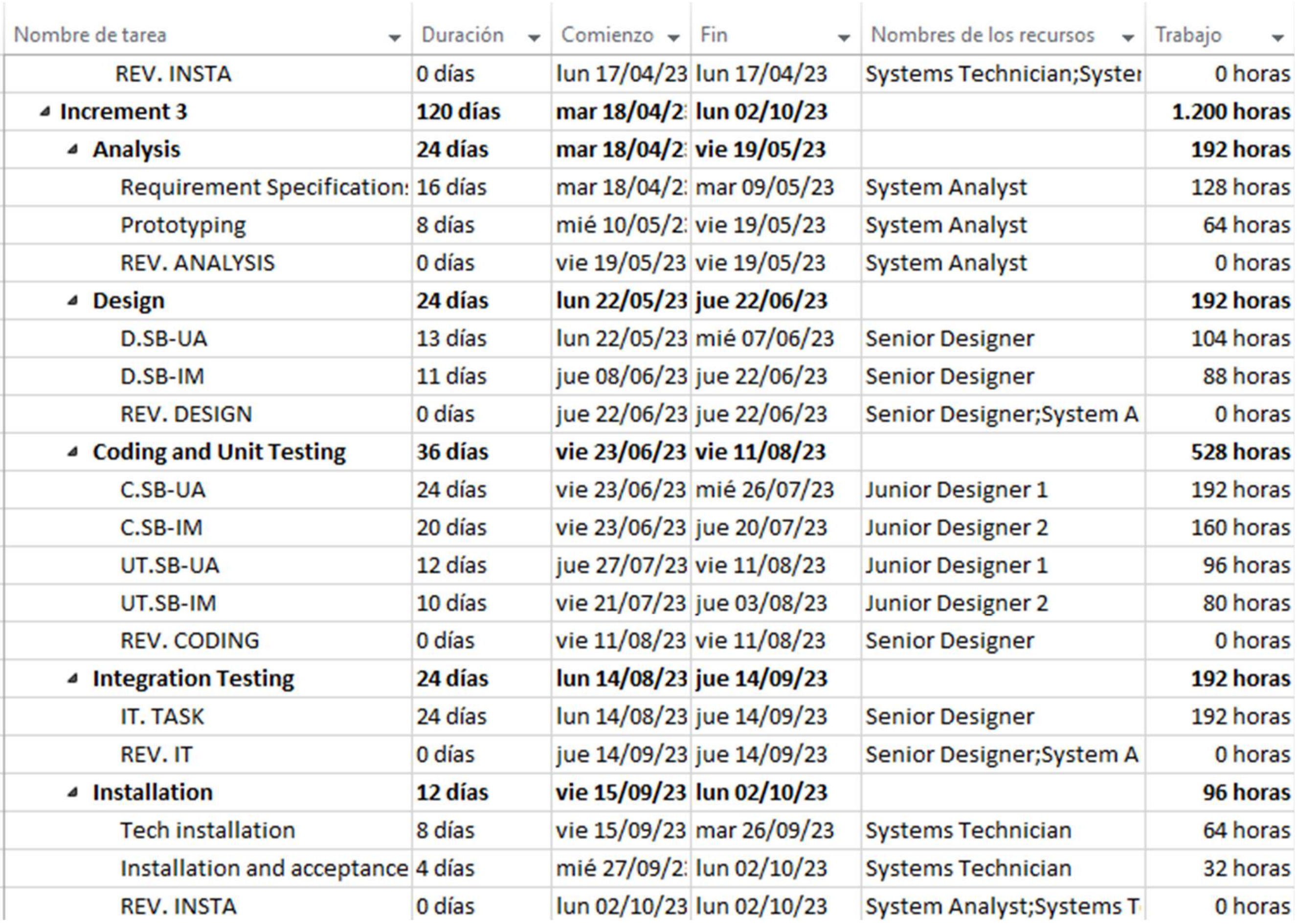
Appendix C: Assigned Qualified Personnel



Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos	Trabajo
Share It! Project	**374 días**	**lun 25/04/22**	**lun 02/10/23**	**Project Manager;Variable**	**9.752 horas**
Increment 1	**100 días**	**lun 25/04/22**	**mar 13/09/22**		**1.040 horas**
Analysis	**20 días**	**lun 25/04/22**	**mar 24/05/22**		**160 horas**
Requirement Specifications	13 días	lun 25/04/22	jue 12/05/22	System Analyst	104 horas
Prototyping	7 días	vie 13/05/22	mar 24/05/22	System Analyst	56 horas
REV. ANALYSIS	0 días	mar 24/05/2:	mar 24/05/22	System Analyst	0 horas
Design	**20 días**	**mié 25/05/2:**	**mar 21/06/22**		**160 horas**
D.SB-UM	9 días	mié 25/05/2:	lun 06/06/22	Senior Designer	72 horas
D.SB-PR	11 días	mar 07/06/2:	mar 21/06/22	Senior Designer	88 horas
REV. DESIGN	0 días	mar 21/06/2:	mar 21/06/22	Senior Designer;System	0 horas
Coding and Unit Testing	**30 días**	**mié 22/06/2:**	**mar 02/08/22**		**480 horas**
C.SB-UM	18 días	mié 22/06/2:	vie 15/07/22	Junior Designer 1	144 horas
C.SB-PR	18 días	mié 22/06/2:	vie 15/07/22	Junior Designer 2	144 horas
UT.SB-UM	12 días	lun 18/07/22	mar 02/08/22	Junior Designer 1	96 horas
UT.SB-PR	12 días	lun 18/07/22	mar 02/08/22	Junior Designer 2	96 horas
REV. CODING	0 días	mar 02/08/2:	mar 02/08/22	Senior Designer	0 horas
Integration Testing	**20 días**	**mié 03/08/2:**	**mar 30/08/22**		**160 horas**
IT. TASK	20 días	mié 03/08/2:	mar 30/08/22	Senior Designer	160 horas
REV. IT	0 días	mar 30/08/2:	mar 30/08/22	Senior Designer;System A	0 horas
Installation	**10 días**	**mié 31/08/2:**	**mar 13/09/22**		**80 horas**
Tech installation	6 días	mié 31/08/2:	mié 07/09/22	Systems Technician	48 horas
Installation and acceptance	4 días	jue 08/09/22	mar 13/09/22	Systems Technician	32 horas
REV. INSTA	0 días	mar 13/09/2:	mar 13/09/22	Systems Technician;Syster	0 horas



Increment 2	**154 días**	**mié 14/09/2:**	**lun 17/04/23**		**1.528 horas**
Analysis	**32 días**	**mié 14/09/2:**	**jue 27/10/22**		**256 horas**
Requirement Specifications	21 días	mié 14/09/2:	mié 12/10/22	System Analyst	168 horas
Prototyping	11 días	jue 13/10/22	jue 27/10/22	System Analyst	88 horas
REV. ANALYSIS	0 días	jue 27/10/22	jue 27/10/22	System Analyst	0 horas
Design	**32 días**	**vie 28/10/22**	**lun 12/12/22**		**256 horas**
D.SB-PM	10 días	vie 28/10/22	jue 10/11/22	Senior Designer	80 horas
D.SB-RSB	22 días	vie 11/11/22	lun 12/12/22	Senior Designer	176 horas
REV. DESIGN	0 días	lun 12/12/22	lun 12/12/22	Senior Designer;System A	0 horas
Coding and Unit Testing	**42 días**	**mar 13/12/2:**	**mié 08/02/23**		**632 horas**
C.SB-PM	22 días	mar 13/12/2:	mié 11/01/23	Junior Designer 1	176 horas
C.SB-RSB	32 días	mar 13/12/2:	mié 25/01/23	Junior Designer 2	256 horas
UT.SB-PM	15 días	jue 12/01/23	mié 01/02/23	Junior Designer 1	120 horas
UT.SB-RSB	10 días	jue 26/01/23	mié 08/02/23	Junior Designer 2	80 horas
REV. CODING	0 días	mié 08/02/2:	mié 08/02/23	Senior Designer	0 horas
Integration Testing	**32 días**	**jue 09/02/23**	**vie 24/03/23**		**256 horas**
IT. TASK	32 días	jue 09/02/23	vie 24/03/23	Senior Designer	256 horas
REV. IT	0 días	vie 24/03/23	vie 24/03/23	Senior Designer;System A	0 horas
Installation	**16 días**	**lun 27/03/23**	**lun 17/04/23**		**128 horas**
Tech installation	11 días	lun 27/03/23	lun 10/04/23	Systems Technician	88 horas
Installation and acceptance	5 días	mar 11/04/2:	lun 17/04/23	Systems Technician	40 horas
REV. INSTA	0 días	lun 17/04/23	lun 17/04/23	Systems Technician;Syster	0 horas



Nombre de tarea	Duración	Comienzo	Fin	Nombres de los recursos	Trabajo
REV. INSTA	0 días	lun 17/04/23	lun 17/04/23	Systems Technician;Syster	0 horas
Increment 3	**120 días**	**mar 18/04/2:**	**lun 02/10/23**		**1.200 horas**
Analysis	**24 días**	**mar 18/04/2:**	**vie 19/05/23**		**192 horas**
Requirement Specification:	16 días	mar 18/04/2:	mar 09/05/23	System Analyst	128 horas
Prototyping	8 días	mié 10/05/2:	vie 19/05/23	System Analyst	64 horas
REV. ANALYSIS	0 días	vie 19/05/23	vie 19/05/23	System Analyst	0 horas
Design	**24 días**	**lun 22/05/23**	**jue 22/06/23**		**192 horas**
D.SB-UA	13 días	lun 22/05/23	mié 07/06/23	Senior Designer	104 horas
D.SB-IM	11 días	jue 08/06/23	jue 22/06/23	Senior Designer	88 horas
REV. DESIGN	0 días	jue 22/06/23	jue 22/06/23	Senior Designer;System A	0 horas
Coding and Unit Testing	**36 días**	**vie 23/06/23**	**vie 11/08/23**		**528 horas**
C.SB-UA	24 días	vie 23/06/23	mié 26/07/23	Junior Designer 1	192 horas
C.SB-IM	20 días	vie 23/06/23	jue 20/07/23	Junior Designer 2	160 horas
UT.SB-UA	12 días	jue 27/07/23	vie 11/08/23	Junior Designer 1	96 horas
UT.SB-IM	10 días	vie 21/07/23	jue 03/08/23	Junior Designer 2	80 horas
REV. CODING	0 días	vie 11/08/23	vie 11/08/23	Senior Designer	0 horas
Integration Testing	**24 días**	**lun 14/08/23**	**jue 14/09/23**		**192 horas**
IT. TASK	24 días	lun 14/08/23	jue 14/09/23	Senior Designer	192 horas
REV. IT	0 días	jue 14/09/23	jue 14/09/23	Senior Designer;System A	0 horas
Installation	**12 días**	**vie 15/09/23**	**lun 02/10/23**		**96 horas**
Tech installation	8 días	vie 15/09/23	mar 26/09/23	Systems Technician	64 horas
Installation and acceptance	4 días	mié 27/09/2:	lun 02/10/23	Systems Technician	32 horas
REV. INSTA	0 días	lun 02/10/23	lun 02/10/23	System Analyst;Systems T	0 horas

Appendix D: Meetings

MEETING ANNOUNCEMENT

From: Juan
To: Rodrigo, Guillermo, Daniel

DATE AND TIME: 08/03/2022 - 13:00
PLACE: UAM
DURATION: 1 hour

PURPOSE: Sections 2 & 3

1. **AGENDA:**
    a. Finish Section2 explanation
    b. Introduction to optional assignment 2
    c. Assignment 2 explanation and guidelines
    d. Start Section3 theory

2. **DECISION FOLLOW-UP**

    None since last class was, we just covered theory

3. **DOCUMENTATION**

    Presentation Slides for sections 2 and 3

    Assignment 2 and optional 2 statements.

MEETING MINUTES

DATE AND TIME: 08/03/2022 - 13:00

PARTICIPANTS: Rodrigo, Daniel, Guillermo, Juan

1. **KEY POINTS DISCUSSED**

   Whether we would make or not optional assignment 2

   We should have better timing regarding the workload of assignment 2

2. **DECISIONS MADE**

   We will be doing optional assignment 2, and more specifically, problem 2.

ACTIONS	RESPONSIBLE PERSON	DEADLINE
Watch the Startup.org movie	Juan, Rodrigo, Guillermo, Daniel	15/03/2022
Optional assignment 2 problem 2 question a	Daniel	15/03/2022

Optional assignment 2 problem 2 question b	Rodrigo	15/03/2022
Optional assignment 2 problem 2 question c and d	Juan	15/03/2022
Optional assignment 2 problem 2 question e	Guillermo	15/03/2022
Call next meeting	Guillermo	09/03/2022

MEETING ANNOUNCEMENT

From: Rodrigo
To: Juan, Daniel, Guillermo

DATE AND TIME: 09/03/2022 - 11:00
PLACE: UAM
DURATION: 2 hours

PURPOSE: Sections 2 & 3

1. **AGENDA:**
    a. Attend to analysis of general system characteristic explanation
    b. Review our objectives for the week

2. **DECISION FOLLOW-UP**

Everything was accomplished.

3. **DOCUMENTATION**

   Problem Slides

MEETING MINUTES

DATE AND TIME: 09/03/2022 - 11:00

PARTICIPANTS: Rodrigo, Daniel, Guillermo, Juan

1. **KEY POINTS DISCUSSED**

   Analysis problem solution

   Extra assignment organization

   Future daily meeting objectives and time

2. **DECISIONS MADE**

   Each one will work in optional 2

   Each member of the team will read assignment 2

ACTIONS	RESPONSIBLE PERSON	DEADLINE
Work in optional assignment	Juan	15/03/2022
Work in optional assignment	Rodrigo	15/03/2022
Work in optional assignment	Daniel	15/03/2022
Work in optional assignment	Guillermo	15/03/2022
Read assignment 2.	All team members	15/03/2022

MEETING ANNOUNCEMENT

From: Juan
To: Rodrigo, Daniel, Guillermo

DATE AND TIME: 22/03/2022 - 13:00
PLACE: UAM
DURATION: 1 hour

PURPOSE: Attend class and organize assignment 2

1. **AGENDA:**
    a. Attend to theoretical explanation by the teacher
    b. Overview the assignment 2
    c. Split the work among the members

2. **DECISION FOLLOW-UP**

   The last two classes the teacher was indisposed so we did not progress that much.

   The optional assignment 2 was delivered on time.

3. **DOCUMENTATION**

   Assignment 2 assessment

   Resources slides from theory

MEETING MINUTES

DATE AND TIME: 22/03/2022 - 13:00

PARTICIPANTS: Rodrigo, Guillermo, Juan

1. **KEY POINTS DISCUSSED**

We discussed how and when to do assignment 2 part A

Meetings must still be recorded and added to the report.

Note: Daniel did not come to class since he was ill.

2. **DECISIONS MADE**

We will follow the recommended pace for the practice and therefore have Part A ready by March 28th. Work was divided between the group, and we will be careful to review Assignment 1 correction when it is uploaded.

ACTIONS	RESPONSIBLE PERSON	DEADLINE
Part A 1rst Subsystem	Guillermo, Daniel	28/03/2022
Part A 2nd Subsystem	Rodrigo, Juan	28/03/2022
Watch for assignment 1 upload	All team members	22/03/2022

MEETING ANNOUNCEMENT

From: Juan
To: Rodrigo, Daniel, Guillermo

DATE AND TIME: 23/03/2022 - 11:00
PLACE: UAM
DURATION: 2 hours

PURPOSE: Attend class

1. **AGENDA:**
   a. Attend to theoretical explanation by the teacher
   b. Practice the usage of Microsoft Project
   c. Overview the assignment 2 Part B

2. **DECISION FOLLOW-UP**

   Each one should progress PartA of assignment 2.

3. **DOCUMENTATION**

   Assignment 2 assessment

   Resources slides from theory

MEETING MINUTES

DATE AND TIME: 23/03/2022 - 11:00

PARTICIPANTS: Rodrigo, Guillermo, Juan, Daniel

1. **KEY POINTS DISCUSSED**

We realized that we were behind with the times and should be starting Part B already.

2. **DECISIONS MADE**

We will do Part A as soon as possible. Part B will be split and finished between March 28th and April 4th.

ACTIONS	RESPONSIBLE PERSON	DEADLINE
Part A 1rst Subsystem	Guillermo, Daniel	28/03/2022
Part A 2nd Subsystem	Rodrigo, Juan	28/03/2022
Watch for assignment 1 upload	All team members	22/03/2022
Organize Part B	All team members	28/03/2022

MEETING ANNOUNCEMENT

From: Rodrigo
To: Juan, Daniel, Guillermo

DATE AND TIME: 29/03/2022 - 13:00

PLACE: UAM
DURATION: 1 hour

PURPOSE: Attend class

1. **AGENDA:**
   a. Attend to theoretical explanation by the teacher.
   b. Work on assignment 2.

2. **DECISION FOLLOW-UP**

   Each one should progress on assignment 2.

3. **DOCUMENTATION**

   Assignment 2 assessment

   Resources slides from theory

MEETING MINUTES

DATE AND TIME: 29/03/2022 - 13:00

PARTICIPANTS: Rodrigo, Guillermo, Juan, Daniel

1. **KEY POINTS DISCUSSED**

   We decided when to meet up and work together on part B of the assignment 2.

2. **DECISIONS MADE**

   We decided to work together on part 2 on March 31st at 5 pm.

ACTIONS	RESPONSIBLE PERSON	DEADLINE
Work on part A and B of assignment 2.	Guillermo, Daniel	31/03/2022
Work on part A and B of assignment 2.	Rodrigo, Juan	31/03/2022

MEETING ANNOUNCEMENT

From: Juan
To: Rodrigo, Daniel, Guillermo

DATE AND TIME: 30/03/2022 - 11:00
PLACE: UAM
DURATION: 2 hours

PURPOSE: Work on P2

1. **AGENDA:**
    a. Work on assignment 2.
    b. Review P1 with the teacher

2. **DECISION FOLLOW-UP**

    Each one finished Part A as expected up to this point.

3. **DOCUMENTATION**

    Assignment 2 assessment

    Resources slides from theory

    P1 feedback

MEETING MINUTES

DATE AND TIME: 30/03/2022 - 11:00

PARTICIPANTS: Rodrigo, Guillermo, Juan, Daniel

1. **KEY POINTS DISCUSSED**

   We went over the feedback for Practice 1 with the teacher, taking note of the changes to be made. We also asked a few questions about theoretical concepts and discussed among the group some ways to tackle Part B.

2. **DECISIONS MADE**

We will keep on working in PartB. In the next meeting we will do the common work discussed and divide the remaining work from the P1 feedback. We will also organize further work during that meeting. Also, the meeting will be at 04:30 pm.

ACTIONS	RESPONSIBLE PERSON	DEADLINE
Work on part B of assignment 2.	Guillermo, Daniel	31/03/2022
Work on part B of assignment 2.	Rodrigo, Juan	31/03/2022

MEETING ANNOUNCEMENT

From: Daniel
To: Rodrigo, Juan, Guillermo

DATE AND TIME: 31/03/2022 - 16:30
PLACE: UAM
DURATION: 2 hours

PURPOSE: Analyze work done and assign new work to be done

1. **AGENDA:**
   a. Check the finished job from the previous week.
   b. Correct together errors made.
   c. Assign together the TDI specifications.
   d. Schedule next week's work to do.

2. **DECISION FOLLOW-UP**

Four of the six subsystems were detailed correctly. User Management Subsystem is not yet done by Guillermo, and Publication of Rides Subsystem is also yet to be done by Daniel.

3. **DOCUMENTATION**

Assignment 2 assessment

Resources slides from theory

P1 feedback

Function Point Table

MEETING MINUTES

DATE AND TIME: 31/03/2022 - 16:30

PARTICIPANTS: Rodrigo, Guillermo, Juan, Daniel

1. KEY POINTS DISCUSSED

We went over the documents of the different subsystems and discussed and corrected all points that were not clear to be correct.

We built all the needed tables for the next work to be done so we can put together the work done without the need of other meeting.

We split the work to be done about the complexity factors explanation between each member of the group.

We split the work to be done about part C of the assignment.

2. DECISIONS MADE

We will finish working on part B of the assignment. The complexity factors will be done separately, by the different members of the team.

ACTIONS	RESPONSIBLE PERSON	DEADLINE
Finish user management subsystem.	Guillermo	05/04/2022
Finish ride booking subsystem.	Daniel	05/04/2022

Work on complexity factors 1-3	Rodrigo	05/04/2022
Work on complexity factors 4-6, 14	Daniel	05/04/2022
Work on complexity factors 7-9, 13	Guillermo	05/04/2022
Work on complexity factors 10-12	Juan	05/04/2022
Work on the increments table	Rodrigo, Daniel, Guillermo, Juan	06/04/2022
Work on the Gantt Chart	Daniel, Guillermo	07/04/2022
Work on the calculations for theoretical planning	Daniel, Guillermo	07/04/2022
Work on the calculations for theoretical budget	Rodrigo, Juan	07/04/2022

MEETING ANNOUNCEMENT

From: Juan
To: Rodrigo, Daniel, Guillermo

DATE AND TIME: 06/04/2022 - 13:00
PLACE: UAM
DURATION: 1 hour

PURPOSE: Identify remaining tasks and attempt to start the report

1. **AGENDA:**
    a. Check on the work made by Daniel and guille
    b. Start the Project planning in MS-Project
    c. Solve doubts with the teacher

2. **DECISION FOLLOW-UP**

All subsystems function points were finished, but the TDIs assigned to Daniel and Guillermo still remain.

3. **DOCUMENTATION**

Assignment 2 assessment

Resources slides from theory

Slides about MS-Project

MEETING MINUTES

DATE AND TIME: 06/04/2022 - 13:00

PARTICIPANTS: Rodrigo, Guillermo, Juan, Daniel

1. **KEY POINTS DISCUSSED**

   Daniel and Guillermo finished their TDI and everyone updated their summary parts. Rodrigo and Juan started creating the MS-Project and defined the resources. The project content was reviewed again in order to ease future assignments of the report.

2. **DECISIONS MADE**

   Daniel And Guillermo will do the project planning table (Gantt Chart). Rodrigo and Juan will advance in the MS-Project file. In the next meeting we will try to divide the report in order to progress during the holidays.

ACTIONS	RESPONSIBLE PERSON	DEADLINE
Work on the MS-Project file	Rodrigo, Juan	07/04/2022
Work on the Gantt Chart	Daniel, Guillermo	07/04/2022
Justify remaining TDIs	Daniel, Guillermo	07/04/2022

MEETING ANNOUNCEMENT

From: Rodrigo
To: Daniel, Juan, Guillermo

DATE AND TIME: 07/04/2022 - 11:00
PLACE: UAM
DURATION: 2 hours

PURPOSE: Analyze work done and assign new work to be done

1. **AGENDA:**
    a. Check the finished job from the previous week.
    b. Correct sheet variations after TDI corrections.
    c. Discuss work partition.
    d. Assign next week's work.

2. **DECISION FOLLOW-UP**

   Up until this point we finished the remaining TDIs, and advanced on the MS-Project file. The Gantt Chart will be created with the MS-Project file.

3. **DOCUMENTATION**

   Assignment 2 assessment

   Resources slides from theory

   Function Point Table

   P1 feedback

MEETING MINUTES

DATE AND TIME: 07/04/2022 - 11:00

PARTICIPANTS: Rodrigo, Guillermo, Juan, Daniel

1. **KEY POINTS DISCUSSED**

   We went over the calculations done in the previous week and revised all numbers where updated after TDI corrections.

   We looked at the non-functional assignments and decided to completely remake them.

   We divided the remaining work to be done, including the SEPRO.

2. **DECISIONS MADE**

   We will finish the MS-Project file, the non-functional assignments, and we will work on the SEPRO document,

ACTIONS	**RESPONSIBLE PERSON**	**DEADLINE**
Finish MS-Project file	Rodrigo & Juan	20/04/2022
Finish non-functional requirements	Guillermo & Daniel	20/04/2022

Finish part 1, 4 & App C of SEPRO	Rodrigo	20/04/2022
Finish part 2, 7 & App D of SEPRO	Guillermo	20/04/2022
Finish part 5 of SEPRO	Daniel	20/04/2022
Finish part 3, App A, App B and summary of SEPRO	Juan	20/04/2022