Software Development Project Management

Date: 21/02/2022 Group: 04

Maciej Cieslinski
Filip Kalinowski
Grant Iodice
Artur Michal Zelik

Client: (name of the company)

Content

1. General data of the company offering the project 6

2. Definitions and acronyms 6

3. Initial offer and budget 7

3.1 Offer 7

3.2 Budget 7

4. Software Configuration Management Plan 8

4.1 Purpose of the Plan 8

4.2 Scope 8

4.3 Definitions and Acronyms 8

4.4 References 9

4.5 Organization 9

4.6 Responsibilities 9

4.7 Applicable policies, directives and procedures 9

4.8 Configuration Identification 10

4.8.1 The preliminary product hierarchy is established 10

4.8.2 Selection of the configuration elements 10

4.8.3 Selection of the identification scheme 11

4.8.4 Definition of relationships 11

4.8.5 Definition and establishment of baselines 12

4.8.6 Definition and establishment of software libraries 12

4.9 Changes control 13

Applicable change control procedure 13

Change Request Report Format 13

Change Certification Report Format 13

4.10 Status account 14

4.11 Configuration auditing 14

5. Quality Plan 14

6. Estimation 27

7. Planning 28

8. Planning and requirements specification 28

8.1 Feasibility study 28

8.1.1 Requirements definition 29

FUNCTIONAL REQUIREMENTS 30

NON-FUNCTIONAL REQUIREMENTS 35

8.1.2 Study of alternative solutions 37

8.1.3 Valuation of alternatives 37

8.1.4 Solution selection 37

8.2 Use case model and traceability matrix 37

8.3 Use cases high level description 39

8.4 Use cases prioritization 42

9. Construction 44

9.1 First Iteration 44

9.1.1 First iteration analysis 44

Expanded format use cases description 44

Operation contracts 48

9.1.2 First iteration Design 50

Sequence diagrams 50
Class Diagram 51
Transition State Diagram 52
9.2 Second Iteration 53
9.2.1 Second iteration analysis 53
Operation contracts 57
9.2.2 Second iteration Design 58
Sequence diagrams 58
Class Diagram 59
Transition State Diagram 60
10. Execution of the quality plan 60
First iteration 61
Analysis 61
Design 61
Second iteration 62
Analysis 62
Design 63
11. Execution of the configuration management plan 63
Configuration Elements Identifiers 63

Tables index

Table 1. Use Cases time estimation **6**
Table 2. Budget estimation **6**
Table 3. Software Libraries **12**
Table 4. Use Cases Prioritization Criteria **40**
Table 5. Use Cases Prioritization Order **41**

Figures index

Figure 1. Use Cases diagrams **7**
Figure 2. General system structure **10**

CREATIVE IDEA DESCRIPTION

Nowadays under covid’s dynamic development, we have to adjust to the new reality of worker burnout. It has become a major problem for many companies across the world. A lot of research shows that nearly half of all workers feel at least somewhat burned out. The modern workforce is quickly becoming languished under current workloads, and are suffering from exhaustion and sleep deprivation as a result. Employees need to meet impossible deadlines and are in constant stress.

Companies, especially larger companies, need a workforce that is not only on the cutting edge, but is the cutting edge. Developing a worker-friendly environment is crucial towards that. To that end, we are developing a new piece of software that will rejuvenate companies’ workforce.

Our idea is to develop an application/website through which employees can book time off from work. This time off can be spent at a rented restroom, in which you can be playing games, or doing some private or group meditation. By earning points based on the number of hours worked, meeting deadlines, and keeping a good efficiency score employees, can spend these points at company-approved spaces for private time and some extra benefits. These break times ensure worker burnout is minimized by de-stressing and detoxing employees.

Company managers can approve worker schedules, and can request modifications from workers if they feel such is necessary. This symbiosis ensures compatibility between worker and management, and all shall prosper in the long run.

A happy workforce is an industrious workforce, and minimizing burnout optimizes output.

Previous attempts at reducing burnout have been initiated by individuals, with no central hierarchy promoting this reduction. By and large, the only source of information to reduce burnout has come from self-help books, articles, and websites.

At the company level, the attempts and resources to reduce burnout have been skimpy. Largely they have only been to “check in” on workers and offer a few measly perks.

Our approach is innovative because it puts the power in employees’ hands. They decide (with company oversight) when to take breaks and what their company rewards are.

1.General data of the company offering the project

- **Name**: JustPower
- **Acronym**: JP
- **Description**: We are a startup company that wants to offer a solution for burnout syndrome.
- **Mission**: Develop an application through which we could enrich breaks type among employees.

2.Definitions and acronyms

- Definitions:
  Burnout - Exhaustion and lack of sleep that comes with excessive workloads.

- Acronyms:
  JP - JustPower

3.Initial offer and budget

Having developed our use cases diagrams, we could predict the duration of our project:
1.5 months

Type of use case	Description	Number of use cases	Time (months)
Simple	1-3 transactions	6	6/4*1=1/2
Average	4-7 transactions	3	¾*1,5=9/8

Table 1. Use Cases time estimation

Total months unadjusted: 1,25
Total months adjusted: 1,5

3.1 Offer

Our offer is to deliver a mobile application for companies whose employees suffer from burnout at their work. The application would enable the users to book time for different activities and enable managers to reward them based on their productivity.

3.2 Budget

There are 6 people developing the project with an equal salary of 2000 € per month. Then we multiply the salary by 1,5 months.

Salary	18000
Computer Equipment	300
Software	150
Consumables	500
Travel & Expenses	200
VAT (21%)	4825.80
Profit (10%)	1915
Risk (10%)	1915
Total	27805.80

Table 2. Budget estimation

Use Cases:

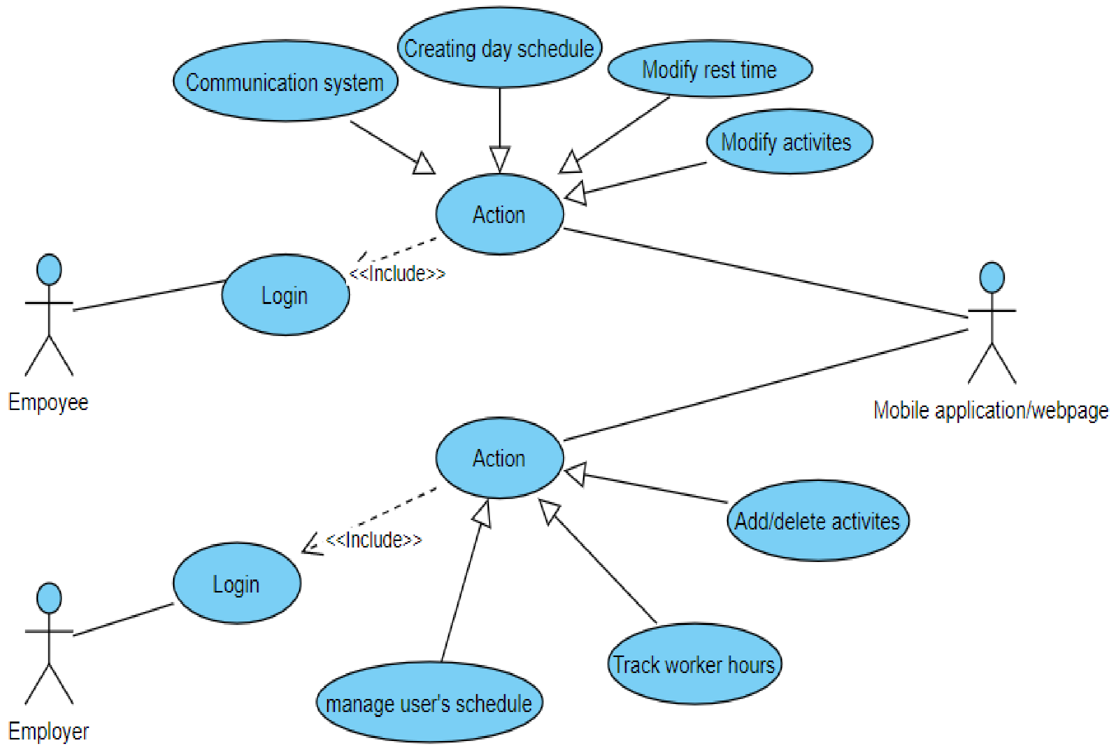


Figure 1. Use Cases diagrams

4.Software Configuration Management Plan

INTRODUCTION

4.1 Purpose of the Plan

This SCM plan documents the basis over which the project stands up firmly and the traceroute to its objective. Its purpose is to track the whole process of the project development – including features addition, changes and reporting of these – in order to successfully achieve the set goals optimally. In the following points, every baseline and change made regarding configuration management activities are gathered in detail.

4.2 Scope

This SCM plan will apply to the **JustPower** project.

4.3 Definitions and Acronyms

Definitions:

- Burnout: the feeling of exhaustion and low motivation that comes with long work hours and lack of sleep.

Acronyms:

- JP: JustPower
- CE: Configuration Element.

4.4 References

"IEEE Standard for Software Configuration Management Plans," in *IEEE Std 828-2005 (Revision of IEEE Std 828-1998)* , vol., no., pp.1-30, 12 Aug. 2005, doi: 10.1109/IEEESTD.2005.96464.

"IEEE Guide to Software Configuration Management," in *ANSI/IEEE Std 1042-1987* , vol., no., pp.1-93, 12 Sept. 1988, doi: 10.1109/IEEESTD.1988.94582.

MANAGEMENT SPECIFICATIONS

This section identifies the coordination and management tasks that will be necessary to carry out the SCM.

4.5 Organization

There must be permanent and direct contact between the development staff and the change control committee, so that delays in the processing of a change are as short as possible, so that both improvement and correction processes are not tedious work.
Both the change control committee and the other development staff should pay special attention to the points where it has been stipulated that baselines will be established within the development. For more information see the section on Definition and Establishment of Baselines**.**

4.6 Responsibilities

Change control committee: **Filip Kalinowski**

Responsible for SCM: ***Grant Iodice***

Librarian: ***Maciej Cieslinski***

Rest of the development staff:

Artur Zelik

David Yael Rodriguez

Ignacio Frutos

Roberto Carlos Vaquerizo

4.7 Applicable policies, directives and procedures

The applicable procedures are described in the section: "Configuration Change Control".

CONFIGURATION MANAGEMENT ACTIVITIES

The following is a description of the SCM activities that will be carried out during the development of this project.

4.8 Configuration Identification

4.8.1 The preliminary product hierarchy is established

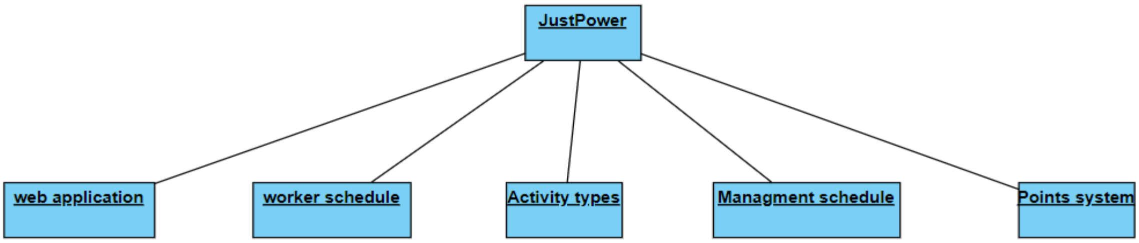


Figure 2. General system structure

4.8.2 Selection of the configuration elements

- Phase 0
    - Offer and budget
    - Quality plan
    - SCM plan
    - SCM plan review
    - First draft of the use case model (if you want to make an estimate based on use case points)
    - Estimation
    - Estimation review
    - Planning
    - Planning review
    - Configuration management plan
    - Configuration management plan review
- Planning and requirements specification phase
    - Feasibility analysis (including requirements specification)
    - Use cases model
    - Prioritization of use cases
    - Definition of high-level use cases
- Construction Phase
    - Analysis
        - Use cases in extended format.
        - Conceptual model
        - Operation Contracts
    - Design
        - Class diagram
        - Sequence diagrams
        - Transition states diagram
    - Coding
    - Testing

4.8.3 Selection of the identification scheme

Identification scheme has two objectives:

1) Provide a unique identifier for each CE
   There are two types of ID:
   - Significant
     Provides additional information as long as it is to remember
   - Non-significant
     Does not provide additional information but it is very easy to assign

2) Defining the information to be stored about CE
   Here are specified information which we want to stored about chosen configuration element:
   - CE code
   - CE name
   - CE description
   - Date of creation
   - Project to which it belongs
   - Baseline to which it belongs
   - Type of CE

The identification scheme is based on a significant identifier model.
The identifiers are divided into two parts following according to scheme AA_BB where AA corresponds to the phase of the development method which is used and BB to the configuration element identified.

This data is stored in the following table:

Configuration element	Identifier
CE code	AA_BB
CE name	
CE description	
Date of creation	
Project to which it belongs	
Baseline to which it belongs	
Type of CE	

4.8.4 Definition of relationships

Our CE’s include: the worker day schedule, the mobile application, and the web application. The mobile app and the web app are related in that they allow both employees and managers to access the same system. The worker day schedule is related to both applications in that it is generated, updated, and managed through our applications.

4.8.5 Definition and establishment of baselines

We might consider baselines to be like our reference points in the software development process. For our project, we will consider the following to be baselines:

- Creating the login page
- Creating the worker day schedule
- Creating the web application
- Creating the mobile application

4.8.6 Definition and establishment of software libraries

Definition:

Software libraries are collections of software and/or related documentation. Their purpose is to assist in the development process and improve the visibility of the system.

Types of libraries:

- ofWork (one per developer)
- Integration (to integrate CEs)
- Support (to house the integrated, and therefore finished, CEs)
- Production (to store the base lines)
- Master (client versions)
- Software or repository (other projects)
- Backup

Establishment of software libraries:

Library	Path location	Control and access protocol
ofWork	server/JP/app/libraries/ofWork/	This column will be specified by a software tool
Support	server/JP/app/libraries/Support/	
Production	server/JP/app/libraries/Production/	
Backup	server/backups/	

Table 3. Software libraries

4.9 Changes control

Applicable change control procedure

Changes in the CEs of the ofWork libraries are informal, as they should not affect any other CEs and at this stage, they aren't part of the baseline.

Changes in the Support and Production libraries are semi-formal. In the support case, the CEs are integrated and interact with each other. In the production case, baselines are stored.

Finally, changes in the master library are formal, as they are definite versions handed to the client.

Change Request Report Format

Requester: needed for contact and feedback

Header of the request: specifies the need for the change to occur and its reasons

Body of the request: describes in detail the changes that need to be made

Affected dependencies: pointers to the items that the approved change would affect, and whether these items form part of another CEs

Risk and opportunities: assessment of potential risks that this change could make in the future, and opportunities to improve the configuration

Change Certification Report Format

Change value: what is thought to be useful for the department or company with respect to this change

Return on investment: whether or not this change will have a positive outcome and when

Change size: how much this change is, i.e. how much affects a determined CE or how many CEs are affected

Complexity: how the change modifies the CE and how its structure is compromised

Available resources: things like human resources, licenses and permits, contractors and freelancers, time, money and knowledge for the change to be possible

4.10 Status account

This information will be completed in section 11.

4.11 Configuration auditing

This information will be completed in section 10.

5.Quality Plan

CONTENT OF THE QUALITY ASSURANCE PLAN FOR THE INFORMATION SYSTEM

In the successive points of the document, the detailed tasks that are going to be carried out in the fulfillment of the Quality Assurance Plan will be exposed to check that the whole project fulfills the necessary quality criteria and that they have been considered as indispensable for the correct accomplishment of the project.

The revisions will be made as the project phases are completed until the final and complete design of the product is reached.

Those responsible for carrying out the revisions and accepting the validity of the products will be Grant as Quality Manager and Yael as Project Manager. In addition, all the members of the work team must carry out the revisions assigned by the Project Manager and communicate to the two people in charge of the Quality Assurance Plan in the event that any fault is found.

The following points of the document detail the specific reviews that will have to be carried out in compliance with the Quality Assurance Plan. The establishment of this quality assurance plan will begin in the System Feasibility Study and will be applied throughout the development of the software project (analysis, design, implementation...).

For each of the revisions, an Audit Report must be added that includes the approval or rejection of the revised product, indicating, if necessary, the causes for rejection of said product.

REVIEW OF THE SYSTEM'S FEASIBILITY STUDY

DOCUMENT REVIEW

Grant, as Quality Manager, will confirm that the requirements have been specified in a structured way, with a precise and complete content, as established in the Quality Assurance Plan. Our Quality Manager will ensure that the requirements specification document offers the following features:

- Identification of absolutely all user requirements.
- Consistency between the content of the document and its objective.
- Each requirement describes the functionality that corresponds to it.
- Correspondence between the requirements of the document and the requirements obtained from the user, so the requirements specification is complete.
- Description of the requirements in clear, unambiguous language and therefore precise
- The feasibility study is self-descriptive, as its structure and content are described.
- A requirements traceability matrix shall be carried out to check that all user requirements have at least one software requirement associated with them and are thus present in the system design.

REVIEW OF USE CASES

REVISION OF THE USE CASE DIAGRAM

Use cases are a very important tool in the software development process and we use them to estimate activities before modeling or building a software development process.

With the use cases we have the functionalities and characteristics or basic requirements of the system. They are not based on any language so they are independent of them.

From the use cases, using the use case method, the size of the software will be estimated. The requirement to be able to use this tool is to define a use case model that represents well the domain of the problem to be addressed.

Artur, as Quality Responsible, must carry out the revision of the Use Case Diagram, for this he must verify that the use case diagram complies with the following:

- The use case diagram describes the behavior of the system, i.e. the complete functionality of the software project to be developed.

- The use case diagram includes all identified use cases representing all system functionalities.

- The use case diagram includes all the actors identified and involved in the system.

- The use case diagram includes all the dependencies and relationships between actors and use cases.

- The use case diagram complies with the graphic notation defined in UML modeling language.

- The use case model includes a glossary of terms that describes the terminology used.

REVIEW OF HIGH-LEVEL USE CASES

Grant, as Quality Manager, must carry out the revision of the high level Use Cases, to do so, he must verify that they comply with the following

- The high-level use cases contain the name, actors, description and type of use case.

- Each use case describes how to achieve a single goal, that is, it describes a feature of the system.

- Each use case contains a textual description of the functionality associated with the appropriate level of detail, including ways in which the intended actors could work with the system. The description will use the language of the end user.

- The use cases do not describe internal system functionality, nor do they explain how it will be implemented. They do not include technical jargon.

- Each use case shows the steps that the actor follows to perform an operation.

- The use cases comply with the graphic notation defined in UML modeling language.

CONFIGURATION MANAGEMENT PLAN REVIEW

CONFIGURATION MANAGEMENT PLAN REVIEW

Grant, as Quality Manager, must carry out the revision of the Configuration Management Plan, to do so he must verify that it complies with the following:

- The project includes a Configuration Management Plan for the control and management of changes in which the activities to be carried out are established that allow the control and management of changes in the project.

- The Configuration Management Plan complies with IEEE Std. 828 - 2005: "IEEE *Standard for Software Configuration Management Plans*" and ANSI/IEEE Std. 1042 - 1987: "*IEEE Guide to Software Configuration Management*".

- The management of the configuration defined in the SCM is carried out during all phases of the software project development, including maintenance and change control.

- The SCM describes a change and version control mechanism that ensures the production of quality software.

- The MTS includes the procedure for generating the necessary documentation for recording and monitoring the changes that occur during the development of the project.

REVIEW OF PROJECT ESTIMATION AND PLANNING

REVISION OF ESTIMATE

When planning a project, an estimate of the cost and human effort required must be obtained. Estimation is one of the crucial activities in the software project management process, necessary for project planning.

Artur, as Quality Responsible, must make the revision of the estimate made for the software development project, for this he must review the following:

- The method used to estimate the effort for the development of the software project uses size-oriented metrics based on points of use cases.

- Before each iteration, verify that the estimate has been made taking into account the use cases included in the estimate.

- The use case points for each of the iterations have been calculated following the procedure established for this estimation method which includes the following steps:

    - Classify each iteration between actor and chaos of use according to its complexity and assign a weight according to it.

    - Calculate the complexity of each use case according to the number of transactions or steps in the case.

    - Calculate the Unadjusted Use Case Points of the iteration.

    - Calculate technical and environmental complexity factors.

    - Calculate Adjusted Use Case Points.

- Once the use case points have been obtained for an iteration, verify that the corresponding effort required to carry them out in that iteration has been calculated from them.

PLANNING REVIEW

Planning is the process of establishing objectives and choosing the means to achieve them. It is essential to carry out an analysis of the project in order to foresee from the beginning and during the development of the project the situations that may arise and to create the necessary conditions to be able to solve them or minimize the consequences that they may have on the development of the project and the achievement of the objectives.
Grant, as Quality Manager, must carry out the revision of the planning made for the software development project, for this he must verify the following:

- A prioritization of use cases to be developed has been carried out and the iterations that will make up the complete development of the software and the use cases included in each of them have been defined.

- An estimation of each iteration has been made based on Use Cases. Based on this estimate, planning will be carried out.

- Before starting an iteration, a planning of the iteration will be done based on the estimation of the effort needed according to the points of use cases.

- The planned planning for the development of the software project will be adapted and updated as the project progresses.

- Planning includes how many people should participate in the project team, what technical skills are needed, when to increase the number of people and who will participate.

- The planning done defines how the team that will work on the software development project will be organized.

- The planning follows the methodology applied to the software development project which is, in this case, incremental iterative based on use cases.

- A Gantt chart is included, representing all the activities to be carried out throughout the project development period. The diagram connects the different activities based on their relationships of precedence and defines the estimated resources and times for each activity.

- The Gantt chart reflects the tasks and key dates, the milestones and the dependency between tasks.

- The quality metrics to be applied to the planning carried out will be

- Speed at which objectives or requirements are completed in each iteration

- Urgency and priority of the completed requirements, to check if there is any misalignment with the project objectives and the organization's strategy.

- Requirements completed in iteration.

- Built-in changes and added requirements on the initial scope of iteration

- Number of requirements completed out of total requirements.

- Deviation of project results from initial planning

- Budget available, budget spent and financial deviation from initial planning.

- Customer satisfaction with regard to the results obtained.

TEST PLAN REVIEW

TEST PLAN REVIEW

Grant, as Quality Manager, must carry out the revision of the Test Plan, for this he must do the following:

- It should be checked that there are rules for carrying out the tests so that it is possible to verify that these tests have been carried out, as well as indicating how to act in the event of differences between the expected result and the result obtained.

- A traceability matrix must be carried out to ensure that there is evidence to verify all software requirements.

REVIEW OF THE PRODUCTS OF THE ANALYSIS PROCESS

REVIEW OF USE CASES IN EXPANDED FORMAT

Artur, as Quality Responsible, must carry out the revision of the Use Cases in expanded format, for this he must do the following:

- From each high-level use case, an expanded use case has been built, in each iteration.

- Each expanded use case is composed of two sections, the header that includes the name, actors, description and type of use case, and the body that describes typical events and alternatives to typical events.

- Expanded use cases define the initiator of the use case.

- The body of the use case consists of two columns describing the actions of the actor and the system responses to them.

REVIEW OF THE CONCEPTUAL MODEL OF THE ANALYSIS

Grant, as Quality Manager, must carry out the revision of the Conceptual Model, for this purpose the following must be verified:

- The analysis model represents the aspects of the problem in a way that is close to the concepts of the problem domain and describes the main characteristics of the system. The analysis model carried out in each of the iterations that make up the project will be validated.

- The conceptual model does not include implementation decisions. It will also be verified that it is independent of the implementation.

- The conceptual model complies with the graphic notation of the UML modeling language. You should also check that the notation has the necessary level of detail to represent the problem, without being overloaded.

- The conceptual model has been made through an object model or class diagram (without methods) that defines the system properties. The entities and the relationships between them have been identified for each iteration.

- The quality metrics to be applied to the conceptual model resulting from the analysis in each iteration are the following:

- Semantic quality: correspondence between the model and the domain, i.e. the model reflects the domain. The validity of the model will be verified, i.e. that all the facts included in the model are correct and relevant to the domain.

- Completeness: the model will be checked to ensure that all facts are correct and relevant to the domain.

- Language quality: the modeling language used to capture the domain is a language that is easy to understand by all participants. The formalization of the language allows the execution of the system.

- Syntactic quality: there is a correspondence between the externalization of the model and the extension of the language in which the model is written.

REVIEW OF OPERATING CONTRACTS

Grant, as Quality Manager, must carry out the revision of the operation contracts that are generated, for this purpose the following must be verified:

- For each case of use, there must be a contract of operation for each action of the actor.

- Each operating contract will consist of the following fields: name, responsibilities, cross references, notes, exceptions, output, pre-conditions and post-conditions.

- Cross-references in the contract shall correspond to references to the requirements defined in the project that are resolved with the use case to which the operation contract belongs.

REVIEW OF THE DESIGN PROCESS PRODUCTS

CLASS DIAGRAM REVIEW

Assessing whether the design obtained meets the required quality level is important in order to know the effectiveness of the processes that have been modeled and whether or not they require great effort for their implementation.

Evaluating design class models by applying metrics allows for the detection of shortcomings and potential improvements from early stages of product development, preventing them from spreading to subsequent phases and enabling the creation of a robust system from its conception.

Artur, as Quality Responsible, will have to carry out the revision of the Class Diagrams, for this he will have to check the following:

- Class diagrams will be made for each iteration with UML and the design will be totally independent of the implementation.

- The comprehensibility of the model or facility with which the class diagram can be understood, the analyzability of the model or facility offered by the class diagram to discover its deficiencies or errors, and the modifiability of the diagram or facility offered by the diagram to make a specified modification, either by error, by a concept not taken into account or by a change in requirements, shall be measured.

- The following metrics will be used to measure the structural complexity of the class diagrams:

  - Number of classes: total number of classes.

  - Number of attributes: total number of attributes.

  - Number of methods: total number of methods.

  - Number of partnerships: total number of partnerships.

  - Number of aggregations: total number of aggregation ratios.

  - Number of dependencies: total number of dependency relationships.

  - Number of generalizations: total number of generalization ratios.

  - Number of generalization hierarchies: total number of generalization hierarchies

    - Number of aggregations: total number of aggregation ratios.
    - WMC: class weighted methods, according to their complexity.
    - Maximum ITL: is the maximum ITL value obtained for each class in a class diagram. For a class within a generalization hierarchy, it is the length of the longest path from the class to the root of the hierarchy.
    - Maximum HAgg: is the maximum HAgg value obtained for each class in the class diagram. For a class within an aggregation hierarchy it is the length of the longest path from the class to the leaves.
- The proposed metrics are highly related both to maintenance time and to the comprehensibility, analyzability and modifiability of the designed class diagram.

REVIEW OF SEQUENCE DIAGRAMS

Grant, as Quality Manager, must carry out the revision of the sequence diagrams generated in the project during the design phase of each iteration, for this purpose the following must be verified:

- For each use case, sequence diagrams have been designed that define both the typical course and the atypical courses of the events defined in them.
- The sequence diagrams show the interaction represented by the sequence of messages between the class instances and actors. The diagrams show instances and events that describe the interaction between the classes.
- Time flows down the diagrams and shows the control flow from one participant to another.
- The UML notation is followed in the definition of the diagrams. The elements included in the sequence diagram are:
    - Name of the sequence diagram.
    - Lifelines for actors and class instances.
    - Messages between instances that define the method that the message calls on the receiving lifeline. In addition, the receiving line is linked to an interface or class.
    - Loops indicate the number of times the loop is executed if known.

REVIEW OF STATE DIAGRAMS

Grant, as Quality Manager, must carry out the revision of the state diagrams generated in the project during the design phase of each iteration, for this purpose the following must be verified:

- The defined state diagrams describe the behavior of the system, with each diagram showing the behavior of a single object during its entire life cycle.
- State diagrams contain states and transitions, and the transitions between them include the corresponding events or actions.
- The state diagram shows all possible states that the object goes through during its life in the application as a result of the events that reach it.
- There is an initial state and a final state and all states represented in the diagram are accessible.

6. Estimation

Adjusted Use Case Points (UCP) = UUCP * TCF * EF		84.889	
Person Hours Mulitiplier (PHM) (Per use case)	* A value of 0 means too risky to proceed	28	hours.use-case
Effort in Person Hours = UCP * PHM (just coding)		2376.892	hours.man
Effort in Person Hours whole project		5942.23	hours.man
Esfuerzo Meses Persona Estimados en el Proyecto		37.1389375	MM
Time estimated using COCOMO II Organic Mode	Tdev=2.5(MM)0.38	9.873592727	Months
Average Team Size (Full Time)	Team Size =MM/Tdev	3.761441101	People
Cost		74277.875	Euros

The difference between real time and estimated time is really huge. Our estimation time was 1.5 month, but as we can see above we need to spend 9.87 months on our project. Now we see that our project is more complex than we expected.

Referring to our budget, our final budget is roughly three times higher than we assumed.

The average team size which we obtained is much smaller. We got 4, but we assumed 6 people.

Putting everything together, our final result is slightly different than we expected.

7.Planning

The schedule will be based on three different phases from which we derived our configuration elements. This way we keep in line with the extended Craig-Larman process.

- Phase 0: Management and Control
- Phase 1: Requirements Specification
- Phase 2: Construction

The project consists of eight use cases that will be developed incrementally. Analyzing them carefully and considering the relationships that link them together, we decided to distribute the work into two different cycles. Each cycle contains 4 use cases. All cycles are of comparable complexity and the amount of effort required is pretty much the same.

Regarding the time and effort needed in order to perform the tasks, we will be mainly focusing on the construction phase, as it is the one that will take longer to complete.

Thus, the time estimation for this latter phase was computed in the following way:

Firstly, we divided the total effort estimation by two (as there are two iterations/cycles) in order to know how long it will take to complete each iteration/cycle.

$$635\ man.hours/2 = 317\ man.hours$$

Secondly, we converted the total amount of hours of effort into days of effort for each worker by dividing the quantity by eight. All members of the team work eight hours per day.

$$317\ man.hours/8hours.day = 40\ man.day$$

Finally, we assigned all tasks to the different workers and that assignment gave us the total amount of time required to complete each task independently.

8.Planning and requirements specification

8.1 Feasibility study

IDENTIFYING THE SCOPE OF THE SYSTEM

Our system will be able to interact with the users and help them plan breaks at work and make them more diverse with various either mindfulness or relaxing activities.

IDENTIFICATION OF STAKEHOLDERS IN THE SYSTEM

Stakeholders are the people who have an impact or are impacted by the system. This includes developers, clients, analysts.

8.1.1 Requirements definition

The requirements are going to be described as follow:

Identifier:	
Name:	
Priority:	**Source:**
Necessity:	
Clarity:	**Verifiability:**
Stability:	
Description:	

- The identification of the requirements will be done in the following way:
    - Identifier: UG-Snnn, where
    - U: indicates that this is a user requirement
    - G: General Requirement
    - S: admits the values:
        - C: Capacity requirement
        - A: Restriction requirement
    - nnn: Consecutive numbers to identify a requirement
- The name field summarizes the requirement
- The priority will have one of the following values:
    - High
    - Medium
    - Low
- The source field can have one of the following values:
    - Customer
    - Analysts
- The necessity field will have one of the following values:
    - High
    - Medium
    - Low
- The clarity field will be assigned one of the following values:
    - High
    - Medium
    - Low

- The verifiability field can have one of the following values:
  - High
  - Medium
  - Low
- Stability describes the duration of the requirement over the life of the software.
- The description field serves to explain the requirement.

FUNCTIONAL REQUIREMENTS

Identifier:UG-C001	
Name: User authentication	
Priority: High	**Source: Customer**
Necessity: High	
Clarity: High	**Verifiability: High**
Stability: High	
Description:	Every user has to provide login and a password in order to login to his account, both of them will be encrypted.

Identifier:UG-C002	
Name: User productivity points calculation	
Priority: Medium	**Source: Analysts**
Necessity: Medium	
Clarity: High	**Verifiability: High**
Stability: High	
Description:	Every account collects points, which are later exchanged for extra activities or benefits.

Identifier:UG-C003	
Name: Schedule keeping	
Priority: Low	**Source: Customer**
Necessity: Medium	
Clarity: High	**Verifiability: Medium**
Stability: High	
Description:	Every user can modify his schedule, plan brakes and sign up for relaxing activities.

Identifier:UG-C004	
Name: new activities	
Priority: Medium	**Source: Customer**
Necessity: Medium	
Clarity: High	**Verifiability: High**
Stability: High	
Description:	Administrator or manager can add new activities which will be available to users.

Identifier:UG-C005	
Name: Setting configuration	
Priority: High	**Source: Customer**
Necessity: High	
Clarity: High	**Verifiability: High**
Stability: High	
Description:	Once logged in, users can change preferences in the app (e.g. dark theme, mute modifications).

Identifier:UG-C006	
Name: adding favorite activities	
Priority: Low	**Source: Customer**
Necessity: Low	
Clarity: High	**Verifiability: Medium**
Stability: High	
Description:	Everyone can add some activities to their own bookmark. It can be done thanks to the heart icon on the screen.

Identifier:UG-C007	
Name: creating a calendar	
Priority: High	**Source: Analysts**
Necessity: High	
Clarity: Medium	**Verifiability: Medium**
Stability: High	
Description:	It shows all of the activities which are currently reserved by users. Users can add their own activities to the calendar in order to increase transparency.

Identifier:UG-C008	
Name: setting default settings	
Priority: High	**Source: Customer**
Necessity: High	
Clarity: Low	**Verifiability: Medium**
Stability: High	
Description:	User can restore all the application settings

Identifier:UG-C009	
Name: Cloud	
Priority: Medium	**Source: Customer**
Necessity: High	
Clarity: Medium	**Verifiability: High**
Stability: High	
Description:	Users can store all their data in the cloud and access it from any device.

Identifier:UG-C010	
Name: Start task action	
Priority: Medium	**Source: Analysts**
Necessity: Medium	
Clarity: Medium	**Verifiability: Medium**
Stability: High	
Description:	This action starts the period of specific activity.

Identifier:UG-C011	
Name: summary calculation	
Priority: Low	**Source: Customer**
Necessity: High	
Clarity: Medium	**Verifiability: Medium**
Stability: High	
Description:	This action enables user to see how much time he spent on work, how many points he gathered and what activities can he book

Identifier:UG-C012	
Name: activity lock	
Priority: Low	**Source: Customer**
Necessity: High	
Clarity: Medium	**Verifiability: Medium**
Stability: High	
Description:	Some of the activities can be locked for specific users so they won't be able to access them.

Identifier:UG-C013	
Name: Deleting activities	
Priority: Medium	**Source: Customer**
Necessity: Low	
Clarity: Medium	**Verifiability: High**
Stability: High	
Description:	Users can delete the activities in case when their expectations are not met. Deleted activities shouldn't appear on the list later.

Identifier:UG-C014	
Name: Setting time	
Priority: Low	**Source: Customer**
Necessity: Low	
Clarity: Medium	**Verifiability: High**
Stability: High	
Description:	Users should be able to change the time of the certain activities and set them up as default.

Identifier:UG-C015	
Name: Notification of the completed activity	
Priority: High	**Source: Analysts**
Necessity: Medium	
Clarity: Medium	**Verifiability: Medium**

Stability: High	
Description:	When completing the avitivity, the system should be able to send notification.

Identifier:UG-C016	
Name: activity spot reservation	
Priority: High	**Source: Analysts**
Necessity: Medium	
Clarity: Medium	**Verifiability: Medium**
Stability: Medium	
Description:	When user books a place for an activity it must be booked for his credentials

Identifier:UG-C017	
Name: reward list access	
Priority: Low	**Source: Customer**
Necessity: Medium	
Clarity: Medium	**Verifiability: Medium**
Stability: Medium	
Description:	Users should be able to see the list of rewards they can potentially get.

Identifier:UG-C018	
Name: showing available activities	
Priority: High	**Source: Customer**
Necessity: Medium	
Clarity: Medium	**Verifiability: Medium**
Stability: High	
Description:	Users should be able to see the list of activities which correspond to their profile

Identifier:UG-C019	
Name: Messaging coworkers	
Priority: Low	**Source: Customer**
Necessity: Medium	

Clarity: Medium	**Verifiability: Medium**
Stability: High	
Description:	Users are able to message other employees whom they are friends with

Identifier:UG-C020	
Name: Adding friends	
Priority: High	**Source: Customer**
Necessity: Medium	
Clarity: Medium	**Verifiability: Medium**
Stability: High	
Description:	Users are able to add other employees within the same company and see their activity

Identifier:UG-C021	
Name: Bug reporting	
Priority: High	**Source: Customer**
Necessity: Medium	
Clarity: Medium	**Verifiability: Medium**
Stability: High	
Description:	Once an user encounter a bug in the application he can report it, so developers can easily solve it.

NON-FUNCTIONAL REQUIREMENTS

Identifier:UG-C022	
Name: activities schedule	
Priority: High	**Source: Analysts**
Necessity: Medium	
Clarity: Medium	**Verifiability: High**
Stability: High	
Description:	Each activity has their own schedule, e.g. some of them may be occupied at some time.

Identifier: UG-C023	
Name: Back-up option	
Priority: High	**Source: Analysts**
Necessity: High	
Clarity: Low	**Verifiability: Medium**
Stability: High	
Description:	This option provides a back-up, which is responsible for storing data and saving

Identifier: UG-C024	
Name:data encryption	
Priority: High	**Source: Analyst**
Necessity: High	
Clarity: High	**Verifiability: High**
Stability:High	
Description:	All user's data should be encrypted

Identifier: UG-C025	
Name: Response time of adding activities	
Priority: Medium	**Source: Analysts**
Necessity: Medium	
Clarity: Medium	**Verifiability: Medium**
Stability:High	
Description:	This is a time between the user's action and the response from the server. Users should have guaranteed usability of the system without delays.

Identifier: UG-C026	
Name: activity identificator	
Priority: High	**Source: Analysts**
Necessity: High	
Clarity: Medium	**Verifiability: Medium**
Stability:High	
Description:	All activities have their own unique ID number, which is assigned by the system.

8.1.2 Study of alternative solutions

The possibility of keeping track of employee's working hours and their direct relationship with weekly objectives achieved was also considered. This alternative considers informing employees about their efficiency throughout the week and, if their performance is high, then letting them book restrooms as soon as they are available.

Regarding the implementation of the system, we thought about using shared calendars for each of the rooms and allowing employees to make a reservation via email. We could give meaningful names to each of the rooms so that they were recognizable for any employee. From our perspective, it is a good idea. If the web application failed for any reason, it would still be possible to book a restroom.

Making use of a hybrid cloud system could also be taken into consideration as it combines the whole infrastructure of a local server or a private cloud with the public one. It is used in some companies and helps to deal with and troubleshoot latency. In case we wanted to take control over private data, we would be able to do it.

8.1.3 Valuation of alternatives

We looked at all alternatives carefully. A system where worker rewards were doled out in accordance to weekly objectives felt too arbitrary as those "objectives" could be anything. We felt that a system where reservations are done via email was too inefficient as it requires one person who receives these emails to constantly be checking for new ones. A self-automated system with managerial oversight was better.

8.1.4 Solution selection

Our solution was iterated before. Workers who clock in a certain number of hours each week can reserve time off in company rest rooms for personal rest and relaxation. The power to construct their schedules is in their own hands.

Company management has the power to approve or deny these schedules, and have minimal oversight over their creation. Due to this approach, we predict workers will be far happier, energized, and more productive.

8.2 Use case model and traceability matrix

Let us describe our use cases and how they relate to our requirements.

	Login	Create Day Schedule	Modify Rest Time	Modify Activities	Manage Worker Schedule	Track Worker Hours (Manag	Add/Delete Available	Communication System

					(Manager)	er)	Activities	
UG-C01	X							
UG-C02		X				X		
UG-C03		X	X	X	X	X		
UG-C04				X			X	
UG-C05			X	X	X			
UG-C06				X			X	
UG-C07		X			X			
UG-C08		X			X			
UG-C09		X			X		X	
UG-C010		X						
UG-C011		X	X		X	X		
UG-C012		X		X	X			
UG-C013		X		X	X		X	
UG-C014		X		X	X	X		
UG-C015			X			X		
UG-C016		X		X				

UG-C017				X		X		
UG-C018				X			X	
UG-C019								X
UG-C020								X
UG-C021								X
UG-C022		X			X			
UG-C023					X			
UG-C024					X			
UG-C025		X			X			
UG-CO26		X		X	X		X	

8.3 Use cases high level description

Identifier:	UC01
Name:	Login
Actors:	Users
Type:	Primary
Description:	
User enters the login credentials to login into the account and get access to the set of actions (e.g. Communication system, Modify activities). In case he does not remember the password he can reset it using his email.	

Identifier:	UC02
Name:	Create Day Schedule
Actors:	Users
Type:	Secondary
Description:	
Every user can choose activities, which he has access to. All of the activities have their own length (in minutes). Users cannot sign up for 2 activities which are scheduled at the same time. Schedules can be chosen by the managers as well.	

Identifier:	UC03
Name:	Modify Rest Time
Actors:	Users
Type:	Secondary
Description:	
Every user has a specific amount of time for a break pre-imposed to be used each day. He can split the time according to his own preferences.	

Identifier:	UC04
Name:	Modify Activities
Actors:	Users
Type:	Primary
Description:	
After the schedule of activities is approved each user can modify it later. He can change the time and activities, but it has to be done one hour before the activity starts. Each action should be accepted by the manager.	

Identifier:	UC05

Name:	Manage Worker Schedule (Manager)
Actors:	Users
Type:	Secondary
Description:	
Managers can modify the schedule of an employee in case he is signed up for too many activities and does not spend enough time on working as well as he can approve or reject modified activities.	

Identifier:	UC06
Name:	Track Worker Hours (Manager)
Actors:	Users
Type:	Primary
Description:	
Managers have the option to track how many hours an employee worked and how many of that they registered to the system.	

Identifier:	UC07
Name:	Add/Delete Available Activities
Actors:	Users
Type:	Primary
Description:	
Users are able to modify the activities they are signed up for. If they are interested in one of them they can add it to their schedule. On the other hand if they are no longer interested in that they can delete it from their schedule.	

Identifier:	UC08
Name:	Communication System
Actors:	Users
Type:	Secondary
Description:	
Users are able to add/delete people from the same company as friends, and send them messages via private or group chats.	

8.4 Use cases prioritization

In order to prioritize the tasks that needed to be accomplished, we established and determined the following prioritization criteria:

	Criteria	Weight
A	Amount of resources required by the project	0.3
B	Impact on architectural design	0.2
C	Number of strategic goals satisfied	0.2
D	Financial opportunities involved	0.1
E	Time-critical or complex procedures	0.1
F	Technological innovation	0.1

Table 4. Use Cases Prioritization Criteria

As it can be observed, criteria A has the highest weight. This is because if there is a lack of resources, this use case and/or the rest could be jeopardized, as the resources are limited.

Criteria B and C have equivalent weights as architectural design impact and strategic goals are still crucial for the overall success of the functionality of the developed product.

These criteria above are deemed essential when aiming to develop a minimum viable product.

Finally, criteria D, E and F have the lowest weight. Starting with the financial opportunities, they aren't as important in a single use case but significant when aggregated overall. With the time-critical procedures, they are not considered as important as the impact on architectural design when taken into account, again, other use cases. As a bonus, technological innovation is added to the weight.

These last criteria must be considered in order to keep making decisions and continue with the process.

UC	A	B	C	D	E	F	Total	Order
Login	8	8	9	6	6	5	7.5	4
Create Day Schedule	10	7	10	7	9	8	8.8	1
Modify Rest Time	7	8	7	5	6	6	6.8	6
Modify Activities	7	8	6	5	5	6	6.5	7
Modify Worker Schedule	7	8	7	5	7	6	6.9	5
Track Worker Hours	9	7	9	10	8	5	8.2	3
Add/Delete Activities	5	8	6	6	6	4	5.9	8
Communication System	10	9	7	7	9	10	8.8	2

Table 5. Use Cases Prioritization Order

Considering the relationships that link some of the use cases together as well as the order of priority that has been given to each of them, we considered appropriate to distribute their development in two cycles:

Cycle 1:

- Create Day Schedule
- Communication System
- Track Worker Hours
- Login

Cycle 2:

- Modify Worker Schedule
- Modify Rest Time
- Modify Activities
- Add/Delete Activities

9.Construction

9.1 First Iteration

9.1.1 First iteration analysis

Expanded format use cases description

Use case	Create Day Schedule
Actor	User
Purpose	To create a daily schedule consisting of available activities.
Overview	Users choose activities that they want to take part in during the day. It is impossible to sign up for 2 activities during the same time, the user needs to choose one. Managers have access to the schedule as well.
Type	Primary
References	UG-C002,UG-C003,UG-C007,UG-C008,UG-C009,UG-C010

<table>
<tr><td>Typical course of events</td><td><table>
<tr><td>Actor</td><td>System</td></tr>
<tr><td>2. Choose activities to perform during the day.</td><td>1. Offer available activities</td></tr>
<tr><td></td><td>3. Set the schedule and give access to employee and manager</td></tr>
</table></td></tr>
<tr><td>Alternative course</td><td>Line 2: Any activity was chosen
Line 3: System backs to the main menu due to any interactions.</td></tr>
</table>

<table>
<tr><td>Use case</td><td>Communication System</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Purpose</td><td>To create a social network by adding or removing friends that you work with.</td></tr>
<tr><td>Overview</td><td>Users have access to social networks, where they can add or remove people and also chat with them.</td></tr>
<tr><td>Type</td><td>Secondary</td></tr>
<tr><td>References</td><td>UG-C019,UG-C020,UG-C021</td></tr>
<tr><td>Typical course of events</td><td><table>
<tr><td>Actor</td><td>System</td></tr>
<tr><td>1.Enter the social network.</td><td>2.Propose persons to add.</td></tr>
<tr><td>3.Confirm specific person to add to networks.</td><td>4.Add that person to friends.</td></tr>
<tr><td>4.Delete specific people from friends.</td><td>5.Ask for confirmation.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td>5. Confirm it</td><td>6.Remove person from group list</td></tr></table></td></tr>
<tr><td>Alternative course</td><td>Line 5: Searched person is unrecognizable, ask to check data. Operation won’t go further, if data won’t be accrued.</td></tr>
</table>

<table>
<tr><td>Use case</td><td>Track Worker Hours</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Purpose</td><td>To measure time how employees spent time on and to check how many of them are registered to the system.</td></tr>
<tr><td>Overview</td><td>Thanks to access this functionality, managers can observe how employees spend time on (e.g. work and activities). Also, there is a possibility to check how many of them are registered at a specific time.</td></tr>
<tr><td>Type</td><td>Primary</td></tr>
<tr><td>References</td><td>UG-C02,UG-C03,UG-C011,UG-C014,UG-C015,UG-C017</td></tr>
<tr><td>Typical course of events</td><td><table><tr><td>Actor</td><td>System</td></tr><tr><td>1.Search for a specific person.</td><td>2.Show actual data about the person asked.</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td>3. Analyze measured time</td><td></td></tr></table></td></tr>
<tr><td>Alternative course</td><td>Line 2: Searched person is unrecognizable, ask to check data. Operation won’t go further, if data won’t be accrued.</td></tr>
</table>

<table>
<tr><td>Use case</td><td>Login</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Purpose</td><td>To log in to the app and to get access to a set of actions.</td></tr>
<tr><td>Overview</td><td>Users can enter the app using login credentials. After accessing, he/she gets access to the set of actions (e.g. create schedule, enter to social network). There is a possibility to remember the password in case he/she doesn’t remember it.</td></tr>
<tr><td>Type</td><td>Primary</td></tr>
<tr><td>References</td><td>UG-C01</td></tr>
<tr><td>Typical course of events</td><td><table><tr><td>Actor</td><td>System</td></tr><tr><td>1.Open app</td><td></td></tr><tr><td>2.Write credentials in</td><td>3.Give access to the app.</td></tr></table></td></tr>
</table>

	mandatory fields and press button ‘login’
Alternative course	Line 3: Given credentials are not correct. Ask for confirmation and propose a login/password reminder.

Operation contracts

Operation:	Create plan for a day
Cross reference:	Use case: Create Day Schedule
Preconditions:	User chooses activities and assigns them to a specific time and presses the confirm button.
Postconditions:	● System checks that any activities are not covered during the same time ● System sets schedule for the user and place it to the calendar

Operation:	Access to social network
Cross reference:	Use case: Communication System
Preconditions:	Users can add or remove people. Also, he can communicate via text chat.
Postconditions:	● System checks that given person

	exist in database ● Next, asked for the confirmation for action ● After confirmation. system add/remove friend to/from database

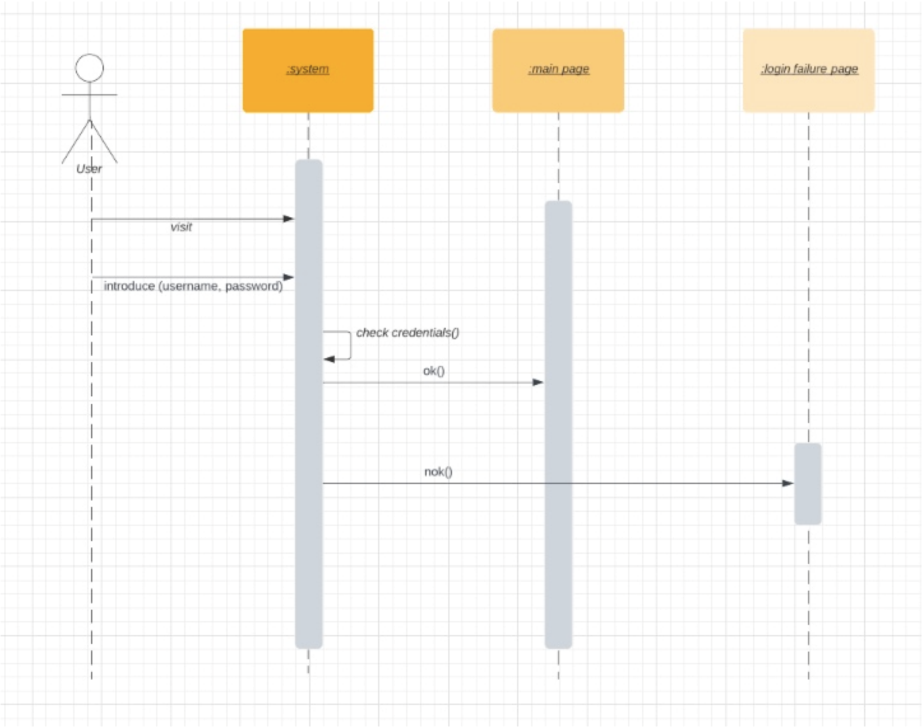
Operation:	Check productivity of employees
Cross reference:	Use case: Track Worker Hours
Preconditions:	Manager access to this functionality and put the date of the employee.
Postconditions:	● System search in database the employee and access to his profile ● Next, system create charts of spent time of employee and present it to the manager (both:schedule and charts of spent time)

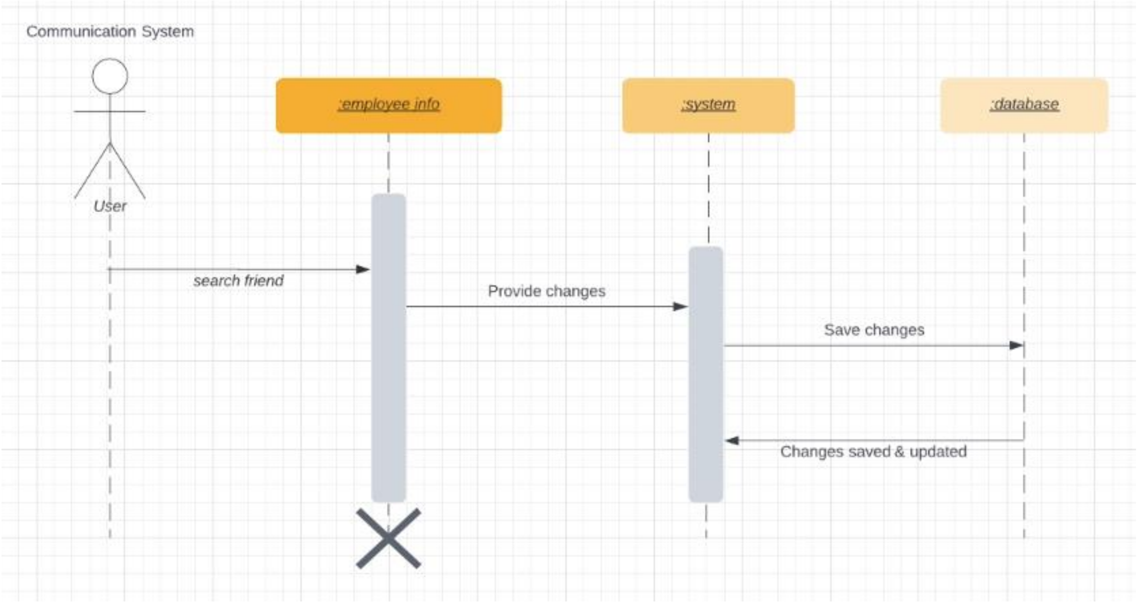
Operation:	Access to an app
Cross reference:	Use case: Login
Preconditions:	Users open the app and put credentials in selected fields.
Postconditions:	● System search for the employee in database and gives access to

	his profile

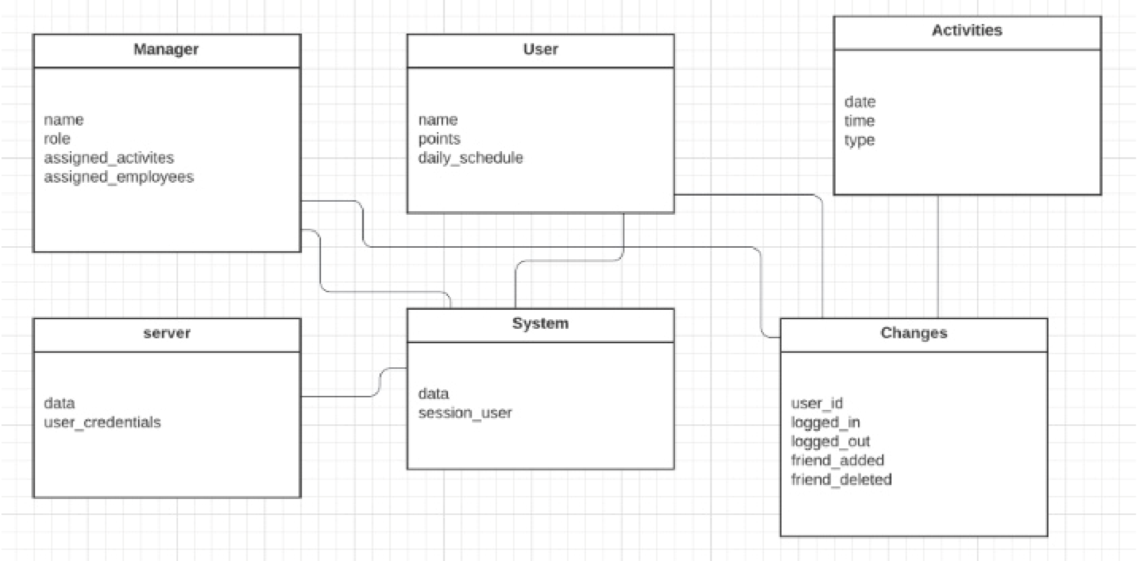
9.1.2 First iteration Design

Sequence diagrams





Class Diagram



For the first iteration, the class diagram conforms to the restroom booking model in which the employee logs in the application, creates tasks on a daily basis and communicates with the other members of the team via the social network.

As can be deduced from the diagram, there are six different classes:

- Manager: this class counts with the name and role of the manager, and two lists containing the activities marked and scheduled for himself/herself as well as the employees who report directly to him/her.

- User: this class is formed by the name, the number of points accumulated depending on the weekly objectives achieved, and the lists of activities marked in the schedule.

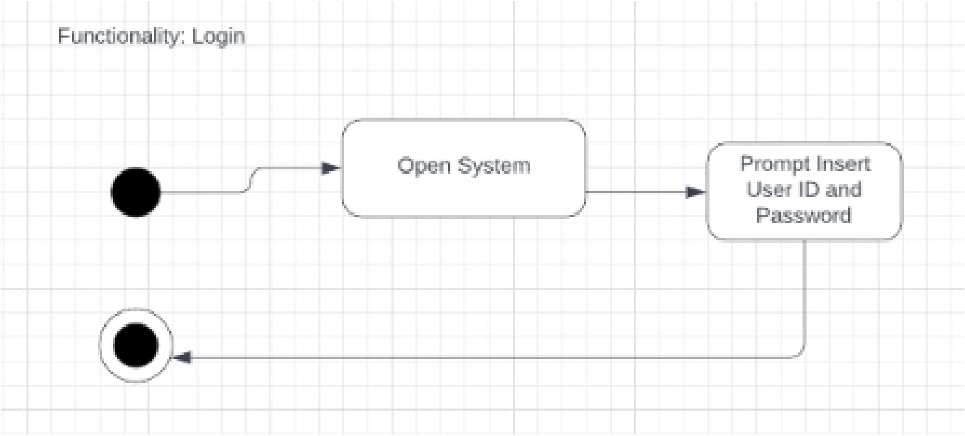
- Activities: this class represents any kind of activities added to the calendar of the employee. Each activity takes variable time and occurs on a specific date.

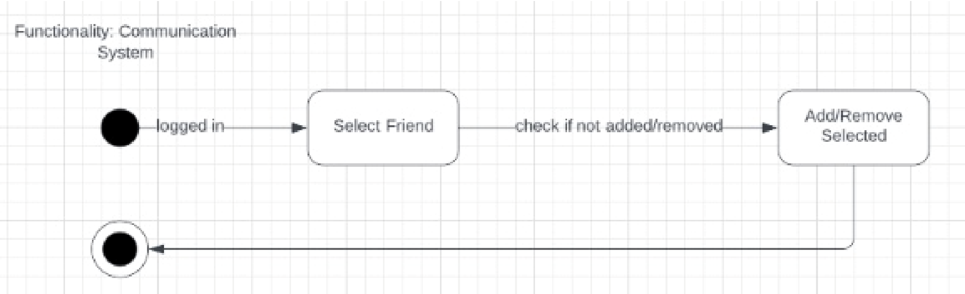
- Server: this class contains all the system data and is connected to the database where the credentials for each employee are stored.

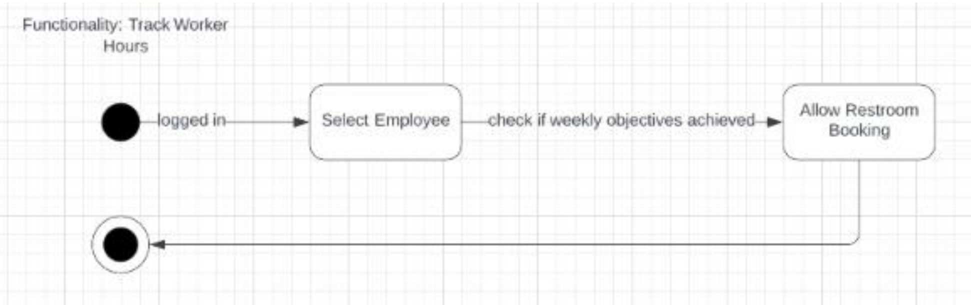
- System: this class keeps temporal data associated with the user session.

- Changes: this class is made of any action that is carried out by the user, either logging in/out the application or communicating with other members of the team.

Transition State Diagram




Functionality: Communication System
logged in
Select Friend
check if not added/removed
Add/Remove Selected


Functionality: Track Worker Hours
logged in
Select Employee
check if weekly objectives achieved
Allow Restroom Booking

9.2 Second Iteration

9.2.1 Second iteration analysis

<table>
<tr><td>Use case</td><td>Modify Worker Schedule</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Purpose</td><td>to estimate the time needed exact tasks</td></tr>
<tr><td>Overview</td><td>Manager can change the worker schedule in order to estimate more efficient work</td></tr>
<tr><td>Type</td><td>Secondary</td></tr>
<tr><td>References</td><td>UG-C03, UG-C05, UG-C07, UG-C08, UG-C09, UG-C11, UG-C012, UG-C013, UG-C014, UG-C022, UG-C023, UG-C024, UG-C025, UG-C026</td></tr>
<tr><td>Typical course of events</td><td>
<table>
<tr><td>Actor</td><td>System</td></tr>
<tr><td>1. Search for a specific worker</td><td>2. Send the current employee schedule to the user</td></tr>
<tr><td>3. Provide some changes</td><td>4. Put into database changes</td></tr>
</table>
</td></tr>
<tr><td>Alternative course</td><td>1. Incorrect name and surname of worker</td></tr>
</table>

<table>
<tr><td>Use case</td><td>Modify Rest Time</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Purpose</td><td>The main purpose of this use case is to provide a way to change the rest time.</td></tr>
<tr><td>Overview</td><td>Every employee can split the time according to their own way.</td></tr>
<tr><td>Type</td><td>Primary</td></tr>
<tr><td>References</td><td>UG-C03, UG-C05, UG-C011, UG-C015</td></tr>
<tr><td>Typical course of events</td><td><table>
<tr><td>Actor</td><td>System</td></tr>
<tr><td>1. log into account</td><td>2. check the logic credential</td></tr>
<tr><td>3. select/change the schedule with your time</td><td>4. approve the schedule in database</td></tr>
</table></td></tr>
<tr><td>Alternative course</td><td>1. incorrect login credential
2. exceed the limit of rest time</td></tr>
</table>

<table>
<tr><td>Use case</td><td>Modify Activities</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Purpose</td><td>providing users with various options where they can change their activities in many ways</td></tr>
<tr><td>Overview</td><td>The user is approved to change the time of the activities, but with some constraints.</td></tr>
<tr><td>Type</td><td>Primary</td></tr>
<tr><td>References</td><td>UG-C03, UG-C04, UG-C05, UG-C06, UG-C012, UG-C013, UG-C014, UG-C016, UG-C017, UG-C018, UG-C026</td></tr>
<tr><td>Typical course of events</td><td><table>
<tr><td>Actor</td><td>System</td></tr>
<tr><td>1. Select the activities, which you want to change and provide changes</td><td>2. Check the time of the activities, respond with message</td></tr>
<tr><td>3. If everything is correct accept changes</td><td>4. Send information to your manager</td></tr>
</table></td></tr>
<tr><td>Alternative course</td><td>2. If some constraints are met block the procedure</td></tr>
</table>

<table>
<tr><td>Use case</td><td>Add/Delete Activities</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Purpose</td><td>Managing our schedule</td></tr>
<tr><td>Overview</td><td>We can freely add or delete our activities. The kind of activities depends on us.</td></tr>
<tr><td>Type</td><td>Primary</td></tr>
<tr><td>References</td><td>UG-C04, UG-C06, UG-C09, UG-C013, UG-C018, UG-C026</td></tr>
<tr><td>Typical course of events</td><td><table>
<tr><td>User</td><td>System</td></tr>
<tr><td>1. Select the activities which you want to add or delete</td><td>2. Check if the activities which you want to add are available.</td></tr>
<tr><td>3. Approve changes</td><td>4. Add/delete activities on the schedule</td></tr>
</table></td></tr>
<tr><td>Alternative course</td><td>2. If the activities is no longer available, send the message to the user</td></tr>
</table>

Operation contracts

Operation:	changing schedule by manager
Cross reference:	Use case: modify Worker Schedule
Preconditions:	Manager can change the schedule using a special account.
Postconditions:	● setting up schedule by the manager in case of some wrongs ● accept changes in database

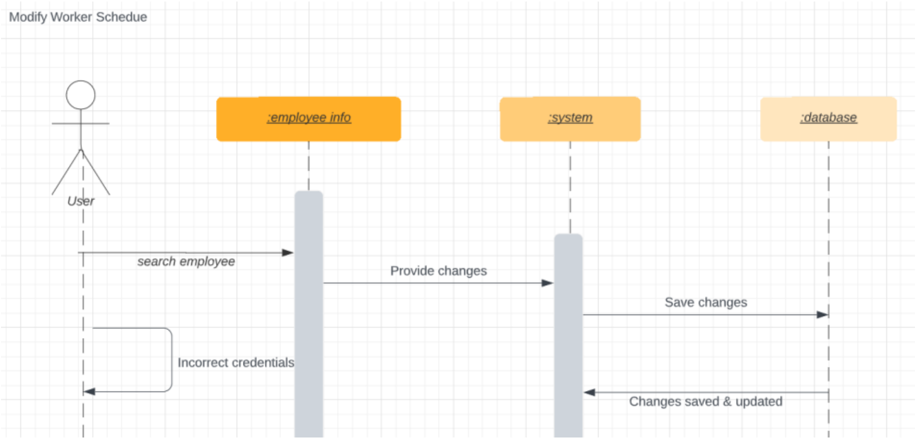
Operation:	save changes of modifying rest time
Cross reference:	Use case: modify Rest Time
Preconditions:	User wants to modify a rest time for his activities. He can do that if he has some free time which can be distributed.
Postconditions:	● Send the changes to the database ● Send information to the manager ● Save and display the changes

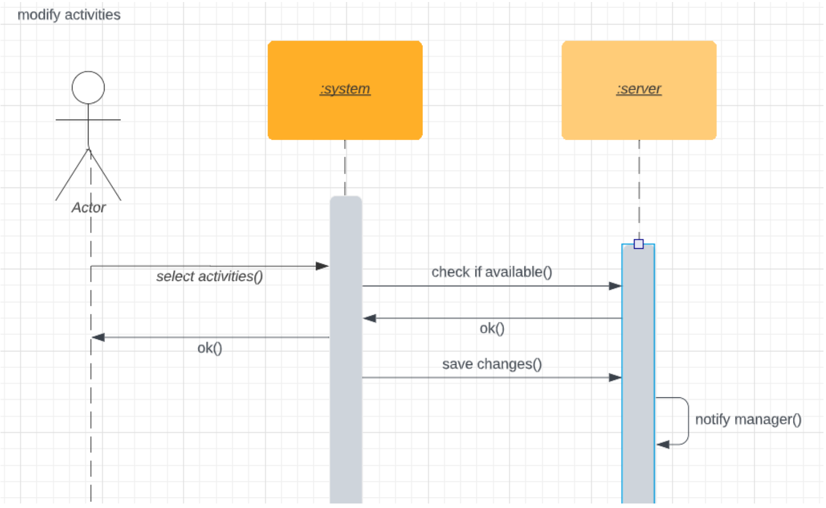
Operation:	changing activities
Cross reference:	Use case: modify Activities
Preconditions:	We want to provide some changes in our schedule, so if we still have time we are able to do that.
Postconditions:	● Send information about it to the manager, who needs to accept the changes. ● Save changes in database

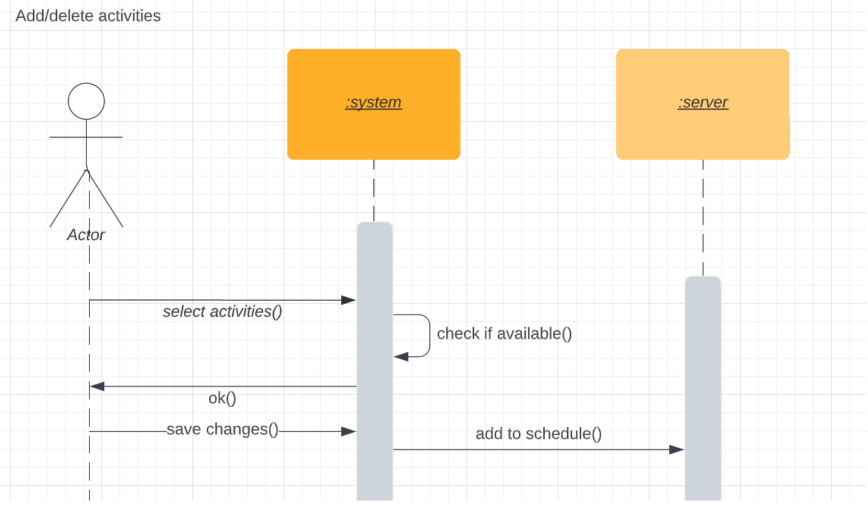
Operation:	Adding new activities
Cross reference:	Use case: add/delete Activities
Preconditions:	The user wants to add new activities, so using the webpage he is able to do that.
Postconditions:	● send information to the database about our changes ● respond with confirmation

9.2.2 Second iteration Design

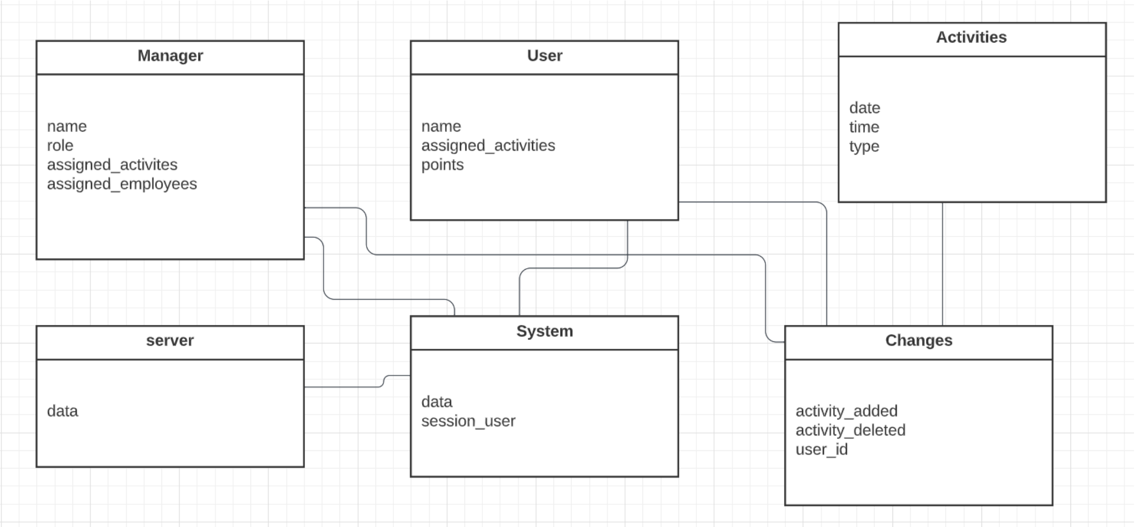
Sequence diagrams



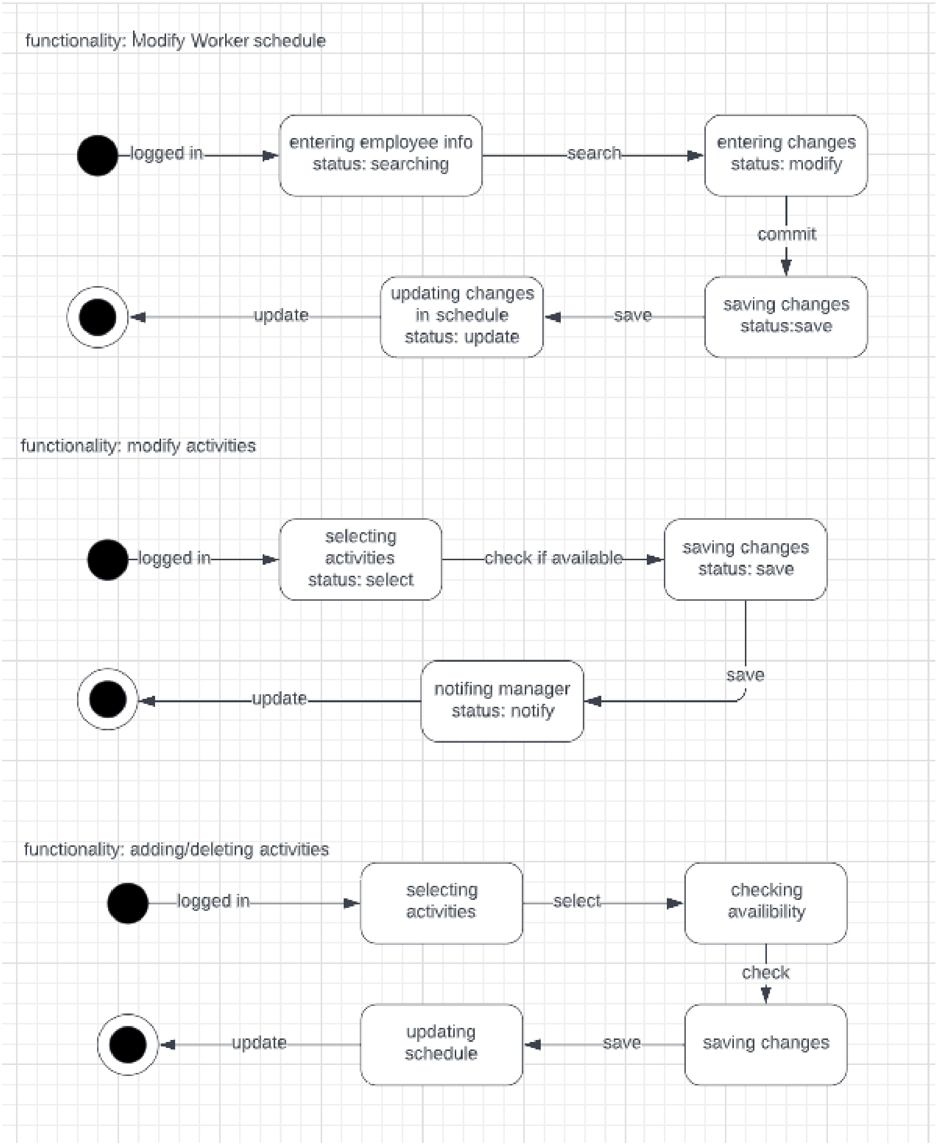




Class Diagram



Transition State Diagram


functionality: Modify Worker schedule
logged in
entering employee info status: searching
search
entering changes status: modify
commit
saving changes status:save
save
updating changes in schedule status: update
update
functionality: modify activities
logged in
selecting activities status: select
check if available
saving changes status: save
save
notifing manager status: notify
update
functionality: adding/deleting activities
logged in
selecting activities
select
checking availibility
check
saving changes
save
updating schedule
update

10. Execution of the quality plan

Here are presented results of execution of the quality plan for each case.

First iteration

Analysis

Document review	Accomplished	date: 09/03/2022
Requirements analysis	Accomplished	date: 14/03/2022
Quality plan	Accomplished	date: 11/03/2022
Use case diagram	Accomplished	date: 23/03/2022
Use case model	Accomplished	date: 31/03/2022
High-level use cases	Accomplished	date: 16/03/2022
Configuration management plan review	Accomplished	date: 16/03/2022
Estimation	Accomplished	date: 18/03/2022
Planning	Accomplished	date: 22/03/2022
Test plan	Accomplished	date: 24/03/2022
Use cases in expanded format	Accomplished	date: 24/03/2022
Conceptual model	Accomplished	date: 29/03/2022
Operating contracts	Accomplished	date: 25/03/2022

Design

Class diagram	Accomplished	date: 14/04/2022

Review of sequence diagrams	Accomplished	date: 11/04/2022
State diagrams	Accomplished	date: 06/04/2022

Second iteration

Analysis

Document review	Accomplished	date: 15/04/2022
Requirements analysis	Accomplished	date: 20/04/2022
Quality plan	Accomplished	date: 20/04/2022
Use case diagram	Accomplished	date: 29/04/2022
Use case model	Accomplished	date: 09/05/2022
High-level use cases	Accomplished	date: 22/04/2022
Configuration management plan review	Accomplished	date: 22/04/2022
Estimation	Accomplished	date: 26/04/2022
Planning	Accomplished	date: 28/04/2022
Test plan	Accomplished	date: 02/05/2022
Use cases in expanded format	Accomplished	date: 02/05/2022
Conceptual model	Accomplished	date: 05/05/2022
Operating contracts	Accomplished	date: 04/05/2022

Design

Class diagram	Pending	date: 23/05/2022
Review of sequence diagrams	Pending	date: 18/05/2022
State diagrams	Pending	date: 13/05/2022

11. Execution of the configuration management plan

Configuration Elements Identifiers

Configuration Elements	Identifier
Offer and budget	P0_OB
Quality Plan	P0_QP
SCM Plan	P0_SCMP
First draft of use case model	P0_FDUC
Estimation	P0_E
Planning	P0_P
Feasibility Analysis	P1_FA
Use cases model	P1_UC
Prioritization of use cases	P1_PUC

Definition of high level use cases	P1_HLUC
Analysis	P2_A
Design	P2_D
Coding	P2_C
Testing	P2_T

Configuration Elements Description

CE Name	Offer and budget
CE Code	P0_OB
CE description	What does our project provide and how much money (and its distribution) will be required
CE date of birth	25/02/2022
CE Type	Document
CE Owner	FILIP KALINOWSKI

CE Name	Quality Plan
CE Code	P0_QP
CE description	Document gathering all the procedures to be followed to ensure quality properties are met
CE date of birth	04/03/2022
CE Type	Document

CE Owner	GRANT IODICE

CE Name	SCM Plan
CE Code	P0_SCMP
CE description	Document tracking evolution of the project – changes, new documents addition, code…
CE date of birth	07/03/2022
CE Type	Document
CE Owner	DAVID YAEL RODRIGUEZ

CE Name	First Draft Use Case Model
CE Code	P0_FDUC
CE description	A provisional design of the Use Case Model
CE date of birth	28/03/2022
CE Type	Document
CE Owner	MACIEJ CIESLINSKI

CE Name	Estimation
CE Code	P1_E
CE description	Approach of the costs involved in the development of the project
CE date of birth	19/04/2022

CE Type	Document
CE Owner	FILIP KALINOWSKI

CE Name	Planning
CE Code	P1_P
CE description	Estimation of the duration of the project using a Gantt Chart
CE date of birth	19/04/2022
CE Type	Document
CE Owner	ROBERTO CARLOS VAQUERIZO ICIERRA

CE Name	Feasibility Analysis
CE Code	P1_FA
CE description	A document where the possibilities of the project to be brought to reality and keep developing it are analyzed
CE date of birth	14/03/2022
CE Type	Document
CE Owner	ARTUR MICHAL ZELIK

CE Name	Use Case Model
CE Code	P1_UC

CE description	A description of how users will use the application developed
CE date of birth	28/03/2022
CE Type	Document
CE Owner	IGNACIO FRUTOS MONEDERO

CE Name	Prioritization of Use Cases
CE Code	P1_PUC
CE description	A guide on how each task may be prioritized based on different criteria
CE date of birth	28/03/2022
CE Type	Document
CE Owner	MACIEJ CIESLINSKI

CE Name	Definition of High Level Use Cases
CE Code	P1_HLUC
CE description	Detailed description of each use case based in each functionality
CE date of birth	28/03/2022
CE Type	Document
CE Owner	ROBERTO CARLOS VAQUERIZO ICIERRA

CE Name	Analysis

CE Code	P2_A
CE description	An inspection of the different elements of the project
CE date of birth	06/05/2022
CE Type	Document
CE Owner	DAVID YAEL RODRIGUEZ SALAMANCA

CE Name	Design
CE Code	P2_D
CE description	Implementation of the components of the project
CE date of birth	06/05/2022
CE Type	Document
CE Owner	GRANT IODICE

Configuration Elements Relationships

Configuration Element Name	P0_FDUC
Related Configuration Elements	P0_OB
Type of Relationship	Derivation
Description	A first draft of the use cases is done prior to the offer and budget

Configuration Element Name	P0_OB

Related Configuration Elements	P0_QP, P0_SCMP
Type of Relationship	Derivation
Description	Quality and SCM plans are concurrently derived from the offer and budget

Configuration Element Name	P0_OB
Related Configuration Elements	P0_E
Type of Relationship	Derivation
Description	An estimation is done after the offer and budget

Configuration Element Name	P0_S
Related Configuration Elements	P0_QP, P0_SCMP, P0_E
Type of Relationship	Composition
Description	The planning is composed of the estimation, quality plan and SCM plan

Configuration Element Name	P0_S
Related Configuration Elements	P1_FA
Type of Relationship	Derivation
Description	The feasibility analysis is done after the scheduling

Configuration Element Name	P0_FDUC
Related Configuration Elements	P1_UC
Type of Relationship	Succession
Description	A first draft of the use cases is followed by the actual definition of the use cases

The following CEs are simple binary derivations listed in chronological order:

Use cases model	P1_UC
Prioritization of use cases	P1_PUC
Definition of high level use cases	P1_HLUC
Analysis	P2_A
Design	P2_D
Coding	P2_C
Testing	P2_T

Baseline Specification

Baseline Name	Phase 0
Baseline Description	Includes the process of formalizing the project previous to analysis and development/production
Baseline date of birth	25/02/2022
Baseline CEs Included	P0_OB, P0_QP, P0_SCMP, P0_FDUC, P0_E, P0_S

Baseline Name	Planning and requirements specification phase
Baseline Description	Feasibility analysis, Use Cases Model and Requirement Specification
Baseline date of birth	14/03/2022
Baseline CEs Included	P1_FA, P1_UC, P1_PUC, P1_HLUC

Baseline Name	Construction Phase
Baseline Description	Software design and software production, including testing
Baseline date of birth	06/05/2022
Baseline CEs Included	P2_A, P2_D, P2_C, P2_T