

Gestión del Conocimiento

uc3m | Universidad Carlos III de Madrid
Grupo 1
Software

Development Project

Date: 22/02/2022 Group: 06
Management

Sean Madsen

Alessandra Priolo

Clement Roger

Lucas Goldman

Margaret Sullivan

Oliver Greenwald

Client: FieldDay
Soren G

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	8
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	14
4.9 Changes control	14
4.10 Status account	15
4.11 Configuration auditing	15
5. Quality Plan	15
6. Estimation	28
7. Planning	29
8. Planning and requirements specification	29
8.1 Feasibility study	29
8.1.1 Requirements definition	29
FUNCTIONAL REQUIREMENTS	29
NON-FUNCTIONAL REQUIREMENTS	31
LEGAL REQUIREMENTS	32
8.1.2 Study of alternative solutions	33
8.1.3 Valuation of alternatives	33
8.1.4 Solution selection	33
8.2 Use case model and traceability matrix	34
8.3 Use cases high level description	34
8.4 Use cases prioritization	36
9. Construction	36
9.1 First Iteration	36
9.1.1 First iteration analysis	36
Expanded format use cases description	36
Operation contracts	40
9.1.2 First iteration Design	49
Sequence diagrams	49
Class Diagram	52

Transition State Diagram	52
9.2 Second Iteration	53
FieldDay only has one iteration, thus this section is not applicable for our project.	53
10. Execution of the quality plan	53
11. Execution of the configuration management plan	54

Tables index

No se encuentran elementos de tabla de ilustraciones.

Figures index

No se encuentran elementos de tabla de ilustraciones.

CREATIVE IDEA DESCRIPTION

Burnout across the world is increasing rapidly leading to detrimental mental and physical effects on employees. Burnout, characterized by low motivation and productivity, can be caused by multiple factors such as stress and lack of engagement and social support. Fortunately, there are ways to combat these causes of burnout in the workplace. Research has shown that employee satisfaction and engagement can be fostered by improving the social environment and sense of shared mission within a company. According to the Harvard Business Review, employees who work in an environment where leadership focuses on engaging employees through positive socialization, regular group communication, and collaboration have a much lower rate of burnout and greater job satisfaction.


To further explain burnout, according to Locke and Latham, accomplishing goals can lead to satisfaction and motivation or could lead to the opposite: frustration, lower productivity, and/or motivation. This is often the cause of burnout when employees feel the latter emotion as a result of not feeling accomplished through their personal and career goals. By providing an environment with fun competition in the workplace, this introduces another goal for employees to focus on and provides more opportunity for them to experience empowering results. Our idea, FieldDay, will specifically address the hierarchy of employee needs starting with building strong social connections and affinity towards the company. Our innovative product will increase the sense of a collective shared purpose among the company, improve the bonds between employees, foster a social support group, create a positive workplace environment, and decrease employee stress levels which accumulate to significantly lower the rate employees experience burnout.

FieldDay will be an app where at the first of each month a game will be revealed with a particular theme. Every month, each employee will play this game once a week, specifically on Wednesdays as a way to break up the work week (although the day of the week can change if need be). Games will be chosen among a group of fun short online games (Wordle, Among Us, Flappy Bird, Doodle Jump, etc.). Throughout the month, the app will present a leaderboard that shows where each employee ranks amongst the company, and where each company ranks among other companies. Companies have the opportunity to incentivize employees to play well by offering fun bonuses to the employees who performed the best that month. Additionally, there will be games, like an online escape room, that involves all the employees on the team working together as a collective unit. Such games will significantly improve the bonds between employees and create a more positive workplace environment. Lastly, some games will be played with partners, with everyone receiving a different partner each

month to ensure everyone in the company gets to know one another. Similar to above, these games will improve the bonds between employees, decrease work stress levels, and ultimately reach the goal of decreasing burnout.

With FieldDay, it is imperative that everybody in the company participates in all the games. From the intern to the CEO. If only a few employees are playing the games, then the connections gained between the employees will be minimal, defeating the purpose of FieldDay. Additionally, it is extremely important that all levels within the company participate to ensure connections are formed between all ranks. Also, managers might believe that FieldDay will decrease productivity, but it is actually the opposite! Taking a little bit of time out of the day to boost social connections within the company and decrease stress levels will increase productivity. Ultimately, FieldDay is an innovative, satisfying, and entertaining way to decrease employee burnout for companies.

1. General data of the company offering the project

- **Name:** FieldDay
- **Acronym:** FD
- **Description:** We are a B2B SAAS company sized 1-10 employees that focuses on bringing together employees from other companies.
- **Mission:** We aim to cultivate relationships between coworkers and bring friendly competition into company cultures.
- **Logo:** The logo for FieldDay, featuring the word "FieldDay" in a bold, sans-serif font. The "Field" part is black and the "Day" part is green. A thin green horizontal line is positioned underneath the "Field" part.

2. Definitions and acronyms

- **Field day:** a day of outdoor sports and athletic competition for school children.
- **Burnout:** Burnout strikes employees when they have exhausted their physical or emotional strength. This usually occurs as a result of prolonged stress or frustration. Sometimes the cause is the work environment. Stressful jobs, lack of support and resources, and tight deadlines can all contribute to burnout.

3. Initial offer and budget

3.1 Offer

The main objective of FieldDay is to provide the necessary software to avoid employee burnout and improve company culture simultaneously at any company. FieldDay is unique due to its innovative way of bringing employees closer together through friendly competition versus other employees and companies. Additionally, the scope of FieldDay is all mid to large sized companies that have roughly 50 or more employees. Specifically, the product is for all employees in the company, not just for managers or developers.

FieldDay will decrease employee burnout throughout the workforce by engaging employees in weekly games and competitions. Once a week, all employees within the company will come together to participate in these games. The games will vary week to week with some being with partners, small groups, or even include the entire company. However, no matter the game, FieldDay ensures that each week employee satisfaction and relationships will improve leading to lower rates of burnout. Ultimately, after analysis, development, and user feedback over the course of a year, FieldDay will be the service that dramatically lowers employee burnout.

3.2 Budget

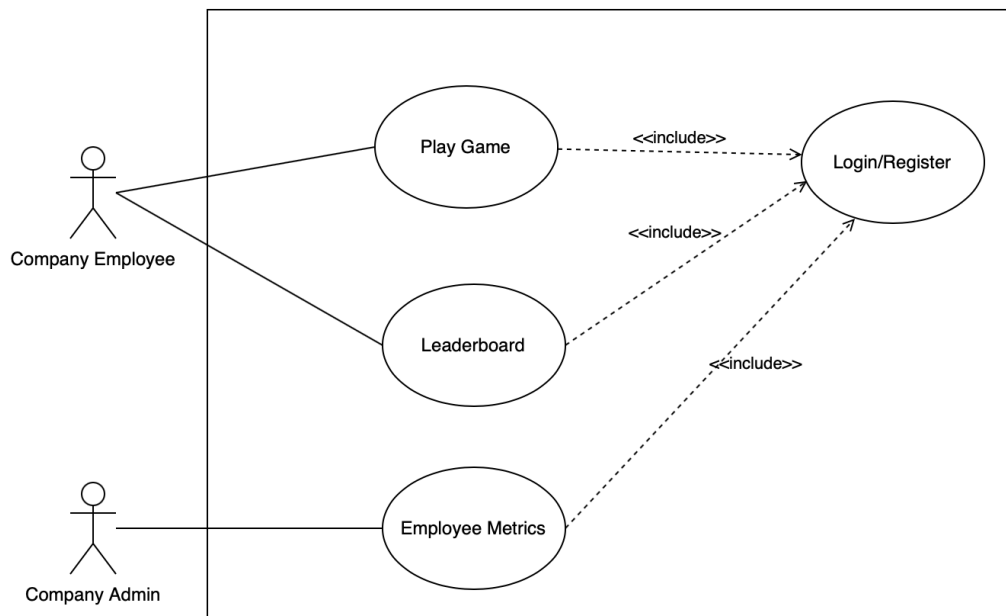
Type of Use Case	Description	Number of Use Cases	Time Management
Simple	1-3 Transactions	0	0
Average	4-7 Transactions	2	$(1/2)*1.5=0.75$
Complex	8+ Transactions	2	$(1/2)*2= 1$

Total Months Unadjusted = $0.5 + 0.5 = 1$ months

Total Months Adjusted = $0.75+1=1.75$ months

To complete this project we have hired 9 employees: one product manager, two psychologists/user experience experts, four backend software engineers, and two frontend developers. Using an average salary of 6,400 Euros a month for project managers, 6400 Euros per month for the psychologists, 8320 Euros of monthly for the backend engineers and 6400 euros per month for the frontend engineers, the total cost is $6,500*1.75 + 6,400*2*1.75 + 8,320*4*1.75 + 6,00*2*1.75 = 113,015$ euros. Thus, to complete the four use cases in 1.75 months, the required budget is 113,015 euros.

Use Case Diagram for FieldDay



4. Software Configuration Management Plan

INTRODUCTION

4.1 Purpose of the Plan

The Plan detailed below is aimed at both the development staff and the management team. The aim is to make the project sufficiently robust to collect information about the state of the product and to make a change. The changes are especially delicate in this one, since there are elements that require special attention and care when modifying them.

It is therefore intended to document each baseline and each change made as indicated below when detailing configuration management activities.

4.2 Scope

This SCM plan will apply to the project FieldDay.

4.3 Definitions and Acronyms

The following are the acronyms used in this Configuration Management Plan.

4.4 References

Coleman, John, and Kristi Hedges. "9 Out of 10 People Are Willing to Earn Less Money to Do More-Meaningful Work." *Harvard Business Review*, 6 Nov. 2018, <https://hbr.org/2018/11/9-out-of-10-people-are-willing-to-earn-less-money-to-do-more-meaningful-work>.

Garton, Eric, and Michael C Mankins. "Engaging Your Employees Is Good, but Don't Stop There." *Harvard Business Review*, 25 Oct. 2017, <https://hbr.org/2015/12/engaging-your-employees-is-good-but-dont-stop-there>.

Lai, Lisa, et al. "4 Reasons Good Employees Lose Their Motivation." *Harvard Business Review*, 10 May 2019, <https://hbr.org/2019/03/4-reasons-good-employees-lose-their-motivation>.

Riopel, Leslie. "The Importance, Benefits, and Value of Goal Setting." *PositivePsychology.com*, 7 Dec. 2021, <https://positivepsychology.com/benefits-goal-setting/>.

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:SS

Alessandra Priolo (Project Manager) and Lucas Goldman (Head Software Engineer)

Responsible for SCM:

Alessandra Priolo (Project Manager)

Librarian:

Clément Roger (Software engineer)

Rest of the development staff:

Sean Madsen (Software Engineer)

Sören Glaser-Gallion (Software Engineer)

Margaret Sullivan (Accountant/Business O)

Oliver Greenwald (Psychologist)

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

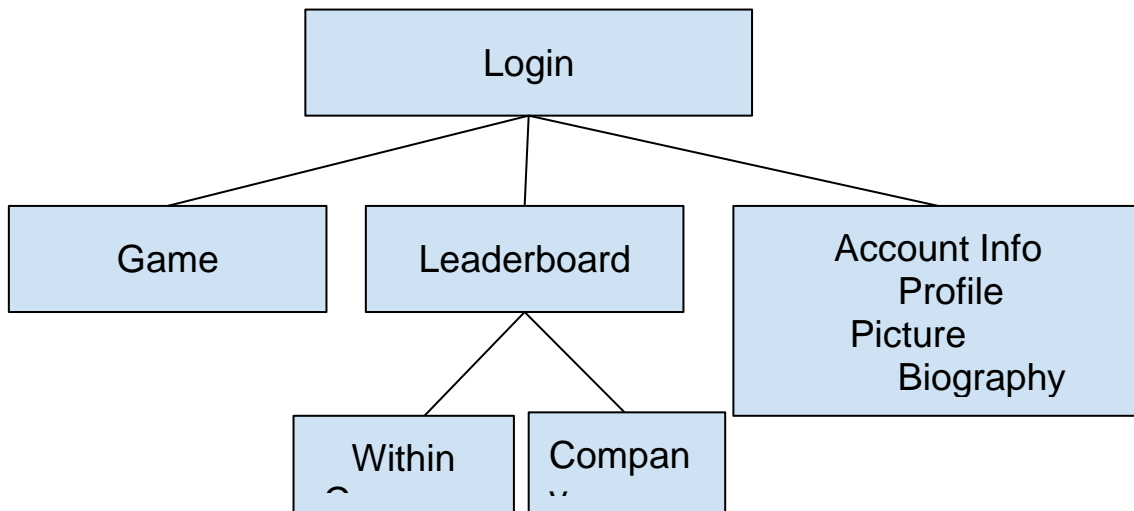


Figure 1 General system structure

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
- Estimation
- Estimation review
- Schedule
- Schedule review

Planning and requirements phase:

- Feasibility analysis
- Feasibility review
- Use cases model
- Use cases model review
- Prioritization of use cases
- Prioritization of use cases review.
- Definition of high-level use cases.
- Definition of high-level use cases review

Construction phase:

- Use cases in extended format

- Use cases in extended format review
- Conceptual model
- Conceptual model review
- Operation Contracts
- Operation Contracts review
- Class diagram
- Class diagram review
- Sequence diagrams
- Sequence diagrams review
- Transition states diagram
- Transition states diagrams review

4.8.3 Selection of the identification scheme

The following fields will be filled out for each individual CE as the project progresses:

- CE code
 - We will use a significant model where the code is xx-yy
 - xx = phase
 - yy = iteration
 - zz = CE acronyms
- Name of the CE
- CE Description
- Date of creation
- Project to which it belongs
- Baseline to which it belongs
- Type of CE (document, program, etc.)

transform into table

each bullet is a column

4.8.4 Definition of relationships

Achieved scores in the game get stored on the Leaderboard with the corresponding username → Game updates Leaderboard

Dependence: There is a dependency between two companies using our software. There needs to be competition and for that to happen there will need to be 2+ companies signed up for our service.

Derivation: The chronological order of events for our project will be the order in which the games are sent out and played. There will need to be decisions made on when we play the games and when they come out in the day for employees to play. Also on a weekly basis in terms of when people can play the first then the second etc. If someone misses a week they will not be able to play a previous week and must follow the order of the games being sent out

Succession: We will have a an automatic portion of code that documents when there is one change of CE to another and store it in an Excel sheet for full viewing by anyone on the team

Structure to portray dependencies: CI-1 and CI-2 mutually depend on one another

CI-1 Code	CI-2 Code	Date
-----------	-----------	------

Structure to portray derivations: CI-1 derives from CI-2

CI-1 Code	CI-2 Code	Date
-----------	-----------	------

Structure to portray succession: Represents the CI-versions along time

CI Code	Previous version	Next version	Date
---------	------------------	--------------	------

4.8.5 Definition and establishment of baselines

- 1) Project scope and definitions are understood by every member of both parties and are agreed upon
- 2) Software company (our company) assigns every role needed to a member of the project and everyone understands their work/role
- 3) Preliminary designs of the code and product are initiated. This will be rapid prototyping and user research/interviews with our target audience. A lot of brainstorming and feedback loops.
- 4) Once a design has been chosen, the project will commence with every team member completing their chosen roles
- 5) The product will be tested with users in a real setting to see how it functions and what feedback they have
- 6) The product is rolled out to the company once enough feedback and iterations have been completed.

Container	Baseline	Iteration
Design Phase 0:	Offer and Budget Quality Plan SCM Plan SCM Plan Review First draft of the use case model Estimation	For each baseline <ul style="list-style-type: none"> ● analysis ● design ● coding ● testing

	Estimation review Schedule Schedule review	
Planning and requirements phase: (Phase 1)	Feasibility analysis Feasibility review Use cases model Use cases model review Prioritization of use cases Prioritization of use cases review. Definition of high-level use cases. Definition of high-level use cases review	For each baseline <ul style="list-style-type: none"> ● analysis ● design ● coding ● testing
Construction phase (Phase 2)	Use cases in extended format Use cases in extended format review Conceptual model Conceptual model review Operation Contracts Operation Contracts review Class diagram Class diagram review Sequence diagrams Sequence diagrams review Transition states diagram Transition states diagrams review	For each baseline <ul style="list-style-type: none"> ● analysis ● design ● coding ● testing

Name of baselines and what will be contained and iterations

Structure to portray baselines:

Baseline name
Baseline code
Baseline description
Baseline date of creation
Baseline CIs

4.8.6 Definition and establishment of software libraries

It is important to establish rules and regulations for control and access so that they do not get lost or damage any code. So for any of the following libraries, the coding team will need to adhere to:

- 1) No coder will have access to libraries until a manager approves a change
- 2) Each library will have comments for every line of code to make sure every coder knows how use each one properly
- 3) Each library will have one central key/passcode when needed that only managers hold and can give out when approved

Libraries used in this project (all will be located in the src folder titled "libraries")

- 1) jQuery
 - a) Helps web pages load faster. Simplifies long lines of code into one method function
- 2) SWFObject
 - a) Used to embed flash SWF files in HTML documents which will be extremely helpful for our games we plan on implementing in our software for employees to compete in
- 3) Three.js
 - a) API that creates beautiful animations which we want and need for our games to "come to life" to make people love our games
- 4) Socket.IO
 - a) Allows for event-based two-way communication between the server and the browser window which will be important for us to communicate game data (i.e. scores) between servers and the browser where we may have games pulled from

4.9 Changes control

The Change Control Committee consists of everyone on the team (refer to section 4.6), with Alessandra Priolo as the leader.

Application Form for Proposed Changes:

1. Why the proposed change is requested:
2. Person who is proposing the change:
3. The exact definition of what needs to be changed:
4. Estimated damage of the change:

5. What areas are affected by the change
6. Change approval:

Once an application has been received, the CCC will decide whether or not to accept the change based on numerous factors such as:

1. Value of the change to the organization:
2. How big the change is:
3. Complexity, cost, and money of the proposed change:

After a decision on the change has been made, the proposed change will start to be worked on. Soon after, the change will be assessed and the originator of the change will be notified of the result.

4.10 Status account

Include here the record of configuration elements, baselines and relationships that occur in this project. (this information can be here or in section 11 of the dossier)

4.11 Configuration auditing

Include here the record of revisions made to configuration items throughout the project, or the management of changes that have occurred throughout the project. (this information can be here or in section 10 of the dossier)

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 FieldDay as Quality Manager and FieldDay 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

Cristina Iniesta Pantoja, 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.

Sören Glaser-Gallion, 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

Cristina Iniesta Pantoja, as Quality Manager, must carry out the revision of the high level Use Cases, to do so, she 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

Cristina Iniesta Pantoja, as Quality Manager, must carry out the revision of the Configuration Management Plan, to do so she 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.

Sören Glaser-Gallion, as Quality Responsible, must make the revision of the estimate made for the software development project, for this she 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.

Cristina Iniesta Pantoja, as Quality Manager, must carry out the revision of the planning made for the software development project, for this she 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

Cristina Iniesta Pantoja, as Quality Manager, must carry out the revision of the Test Plan, for this she 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

Sören Glaser-Gallion, as Quality Responsible, must carry out the revision of the Use Cases in expanded format, for this she 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

Cristina Iniesta Pantoja, 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

Cristina Iniesta Pantoja, 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.

Sören Glaser-Gallion, as Quality Responsible, will have to carry out the revision of the Class Diagrams, for this she 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

Cristina Iniesta Pantoja, 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

Cristina Iniesta Pantoja, 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

Attached below are screenshots from the excel spreadsheet containing information about FieldDay's estimation.

Unadjusted Actor Weighting Table	Description	Weight	Number	Value
Simple	External System with well-defined API	1	1	1
Average	External System using a protocol-based interface, e.g., HTTP, TCT/IP, or a database	2	2	4
Complex	Human	3	2	6
Unadjusted Actor Weight Total (UAW)				11

Above is the unadjusted actor weighting table, with the numbers from the use cases.

Description	Weight	Number	Value
1 – 3 transactions	5	0	0
4 – 7 transactions	10	2	20
> 7 transactions	15	2	30
Unadjusted Use Case Weight Total (UUCW)			50

Additionally, the above figure displays the technical complexity Factors.

Phase	Percentage
Analysis	15,00%
Design	25,00%
Coding	35,00%
Test	15,00%
Extra (other activities)	10,00%

Also, it is worth pointing out that the coding phase is given the highest percentage as that will be the hardest and most time consuming phase.

Cost		20636,10938	Euros
Hours worked per month	116		
Average Monthly Salary (euros)	1031		

Finally, most importantly, the figure above displays the estimated cost of FieldDay with the hours worked per month and average monthly salary.

7.Planning

The Gaant chart specifies the order of the completion of the CI's. We included all CI's listed in section 11, using the same order and duration as specified in section 11. Importantly, we elected to have the product manager complete each CI, as we are a startup we do not have many team members, thus, the product manager will also be doing that which an analyst and quality manager might usually complete.

8.Planning and requirements specification

8.1Feasibility study

The scope of the following section applies to the FieldDay project, specifically the FieldDay website. The stakeholders and responsibilities of the project can be seen in section 4.6.

8.1.1 Requirements definition

FUNCTIONAL REQUIREMENTS

Identifier: UG-S001	
Name: Create User Account	
Priority: High	Source: Customer

Necessity: High	
Clarity: Medium	Verifiability: High
Stability: High	
Description:	Construct the software to create a new user on the FieldDay website and store their account information (name, email, password, company.) into the database. Additionally, create the software for once a user has an account, they can reliably log into the website and recreate their password if need be through email.

Identifier: UG-S002	
Name: Game Leaderboards	
Priority: High	Source: Customer
Necessity: High	
Clarity: Medium	Verifiability: High
Stability: High	
Description:	Code the software such that when a user is logged into their account, they are able to view the results of prior completed games, and how their results compare to their coworkers. Additionally, this involves the software of comparing the results of games between companies.

Identifier: UG-S003	
Name: Server Setup	
Priority: High	Source: Analysts
Necessity: High	
Clarity: Medium	Verifiability: High
Stability: High	
Description:	Construct the underlying hardware for the FieldDay website servers such that users can access the website 24/7 with low latency.

Identifier: UG-S004	
Name: Play Game and Store Results	
Priority: High	Source: Customer
Necessity: High	
Clarity: Medium	Verifiability: High
Stability: High	
Description:	Create the software such that when a user is logged into the FieldDay website, the user can play the weekly game on the website. This involves the process of pairing employees together for games if the game requires a partner or is team based. Additionally, this involves the process of storing the games results into the database for future use (i.e leaderboards).

NON-FUNCTIONAL REQUIREMENTS

Identifier: UG-S005	
Name: Account Security	
Priority: Medium	Source: Analyst
Necessity: High	
Clarity: Medium	Verifiability: Medium
Stability: High	
Description:	Set up the necessary software (and hardware if need be) such that users are only allowed to log into their account and view only their account credentials. Make it such that it is impossible for others to access other users' account information. Users passwords need to be hashed and make use of cryptographic techniques.

Identifier: UG-S006	
Name: Game Performance	
Priority: Medium	Source: Customer
Necessity: Medium	
Clarity: High	Verifiability: High
Stability: Long (high)	

Description:	When a user is playing in the weekly game, the user should experience no performance issues that impact their ability to complete the game. This means that the game should always be running at a minimum of 30 frames per second and 720p.
---------------------	--

LEGAL REQUIREMENTS

Identifier: UG-S007	
Name: Rights to use Games	
Priority: High	Source: Analyst
Necessity: High	
Clarity: Medium	Verifiability: High
Stability: Long (high)	
Description:	Since FieldDay is not constructing most of the games themselves, proper documentation (i.e. a contract between FieldDay and another company) must be required to allow FieldDay to use outside games legally.

8.1.2 Study of alternative solutions

Our software is a new idea that pulls from multiple already existing platforms such as game engines and utilizes user data from a network of companies. Since our software is unique, the alternative solutions are limited to merging existing softwares or building off of existing platforms.

These alternative solutions include:

1. merge our application that holds the company users and data with existing games to make a coherent platform
2. Adopt out existing game applications and redevelop them to fit our platform that would have additional features that are tailored towards our mission
3. Merge existing game applications with systems that hold data from many different companies to easily onboard multiple large companies onto our system when we first launch the product
4. Design a brand new platform with user capabilities and new monthly games for user to play

8.1.3 Valuation of alternatives

Alternative 1: This solution seems viable since we would keep the actual game and the rest of the software separate so they are not dependent on each other to avoid dependency issues. Our software would only use information of the scores of the games and would handle user information and statistics separately from the games.

Alternative 2: This alternative is a good plan, but there may be issues with user rights to play the game and incompatibility of our leaderboard and user accounts with the games.

Alternative 3: This solution may be hard to implement, given that many large scale games may not track much of the game players data, and this data may be held in different formats, which would make it hard to combine this data.

Alternative 4: This solution is not cost effective since there are already many viable games that exist and the cost of developing new games without bugs and other issues takes lots of trial and error. In addition, the development of this would take much longer than utilizing already functional games.

8.1.4 Solution selection

The selected solution is solution 1 since it is the most cost effective and will utilize games that already exist and integrate them into our platform. The features that we will develop are user accounts and a database that holds user information, leaderboards and awards and a schedule of when the games will be played and which game will be played.

8.2 Use case model and traceability matrix

Requirements	Use case 1 (play game)	Use case 2 (Register/Login)	Use case 3 (Leaderboard)	Use case 4 (Employee Metrics)
R1	x	x	x	x
R2	x		x	
R3	x	x	x	x
R4	x			
R5		x		x
R6	x			
R7	x			

8.3 Use cases high level description

Use Case:	Play Game
Actors:	Client/Company Employee
Type:	Primary
Description:	The client (company employee) navigates to the weekly FieldDay game from the FieldDay homepage. Ensuring that the client is logged in, the client is assigned their partner or teammates, while also viewing other partners and teams. Next, the client plays the assigned game where the results are stored on the database and displayed on the leaderboard. Lastly, the client's manager is notified that the client played their game.

Use Case:	Register Account
Actors:	Client/Company Employee
Type:	Primary
Description:	The client (company employee) navigates to the FieldDay register page from the FieldDay homepage. Once there, the client fills out their user information (name, company, email, password, etc) and creates their account. The inputted information is stored in the FieldDay database and the client is sent to the login page. On the login page, the user can login into their account, or if the client forgot their password they can reset their password with their email.

Use Case:	Display Leaderboards
Actors:	Client/Company Employee
Type:	Primary
Description:	The client (company employee) navigates to the FieldDay leaderboard page from the FieldDay homepage. Ensuring that the user is logged into their account, the client can either view the company leaderboard or the global leaderboard broken down into small, medium, and large companies. Also, for both leaderboards, the client can sort the results by weekly, monthly, and all-time scores for the games.

Use Case:	Consult Employee Metrics
Actors:	Client/Company Admin
Type:	Secondary
Description:	The client (company admin) navigates to the FieldDay employee metrics page from the FieldDay homepage. Ensuring that the client is logged in, specifically a manager, the client can view the results of all the games of their employees. Additionally, the client can reach out to employees that are not participating in the games. Finally, based off the results of the games the client can specify rewards for specific employees and let the winners know they won.

8.4 Use cases prioritization

Weighting	0.2	0	0.3	0	0.3	0.2	
Use Case	a	b	c	d	e	f	Sum:
Play Game	3	-	4	-	5	2	3.7
Register Account	4	-	5	-	2	1	3.1
Display Leaderboards	1	-	2	-	3	1	1.9
Employee Metrics	2	-	3	-	5	4	3.6

Our Development will only require a single cycle since we only have four use cases.

First Cycle:

Register Account

Play Game

Employee Metrics

Display Leaderboards

With the above prioritization, it can be observed that in each cycle more functionality is provided to the system, expanding from the general functionalities and ending with the more specific niche functions of our application including the leaderboard and employee metrics.

9. Construction

9.1 First Iteration

9.1.1 First iteration analysis

Expanded format use cases description

Use Case:	Login/Register
Actors:	Employee or Company Manager (initiator)

Purpose:	To create an account for a user, or instantiate session with the user's credentials
Overview:	The client (company employee) navigates to the FieldDay register page from the FieldDay homepage. Once there, the client fills out their user information (name, company, email, password, etc) and creates their account. The inputted information is stored in the FieldDay database and the client is sent to the login page. On the login page, the user can login into their account, or if the client forgot their password they can reset their password with their email.
Type:	Primary
References:	UG-S001, UG-S005
Typical Course of Events:	<ol style="list-style-type: none"> 1. Use Case begins with user navigating to the login/register page on the FieldDay website 2. If the user has an account already: <ol style="list-style-type: none"> a. User inputs username b. User inputs password c. System verifies that the entered data is valid, the user is logged in. Otherwise, the user is asked to re-enter information. 3. If the user does not have an account: <ol style="list-style-type: none"> a. User inputs identifying details like name and email b. User inputs username c. User inputs password d. User clicks to register account e. Information is registered with FieldDay databases and user is logged in
Alternative Courses:	Line 2: If the user already has an account but can't remember their password, they have the option to reset their password.

Use Case:	Play Game
Actors:	Client/Company Employee (Initiator)
Purpose:	To play the weekly game for each employee
Overview:	The client (company employee) navigates to the weekly FieldDay game from the FieldDay homepage. Ensuring that the client is logged in, the client is assigned their partner or teammates, while also viewing other partners and teams. Next, the client plays the assigned game where the results are stored on the database and displayed on the leaderboard. Lastly, the client's

	manager is notified that the client played their game.
Type:	Primary
References:	UG-S001, UG-S003, UG-S004
Typical Course of Events:	<ol style="list-style-type: none"> 1. User navigates to the 'Play Game' webpage on the FieldDay website (ensuring that user is logged in) 2. User is partnered up with other employees throughout the company 3. User plays the game until completion 4. The results of the game session are stored in the FieldDay databases
Alternative Courses:	Line 2: For some games they will be completed individually, thus there is no need to partner up employees.

Use Case:	Leaderboard
Actors:	Client/Company Employee (Initiator)
Purpose:	To display the game sessions of employee's company
Overview:	The client (company employee) navigates to the FieldDay leaderboard page from the FieldDay homepage. Ensuring that the user is logged into their account, the client can either view the company leaderboard or the global leaderboard broken down into small, medium, and large companies. Also, for both leaderboards, the client can sort the results by weekly, monthly, and all-time scores for the games.
Type:	Primary
References:	UG-S002, UG-S004
Typical Course of Events:	<ol style="list-style-type: none"> 1. User Navigates to the 'Display Leaderboard' webpage on the FieldDay website (ensuring that the user is logged in) 2. System gathers and calculated all of the game sessions for the company that the user works for for that week 3. User is displayed all of the game results for the game that was completed that week
Alternative Courses:	Line 3: User can decide to either view the game results for just that week, or game results from the past month or all-time scores

Use Case:	Employee Metrics
Actors:	Client/Company Admin (Initiator)
Purpose:	To compute trends and statistics for employees based off game session data
Overview:	The client (company admin) navigates to the FieldDay employee metrics page from the FieldDay homepage. Ensuring that the client is logged in, specifically a manager, the client can view the results of all the games of their employees. Additionally, the client can reach out to employees that are not participating in the games. Finally, based off the results of the games the client can specify rewards for specific employees and let the winners know they won.
Type:	Secondary
References:	UG-S004, UG-S001
Typical Course of Events:	<ol style="list-style-type: none"> 1. User navigates to the 'Employee Metrics' webpage on the FieldDay website 2. System verifies that the user logged in is a manager or someone with privileged access 3. User sets filters to decide which employees they want to analyze 4. System gathers all of the necessary information about employee's game metrics and calculates trends, patterns, etc. about specified employees 5. User (who is most likely a team manager) analyzes the game session results for their employees and takes note of any concerning or interesting trends 6. User requests to talk to a specific employee if the user notices something worrying
Alternative Courses:	<p>Line 5: Sometimes the user will only want to see results for a specific employee, not multiple such as all interns</p> <p>Line 6: All employees are doing well and there's no need to talk to a specific employee.</p>

Use Case: Play Game	
Name: RetrieveUserID(CurrentUserID)	
<i>Responsibilities</i>	<i>Return the ID number of the current user based on their login credentials</i>
<i>Cross-References</i>	<i>Functional Requirements: UG-S001, UG-S003, UG-S005 Use Cases: Play Game</i>
<i>Notes</i>	<i>For each login credential (login and matching password) there can be only one UserID affiliated with it.</i>
<i>Exceptions</i>	<i>If there is no user that corresponds with the credentials, indicate that there has been an error.</i>
<i>Output</i>	<i>If a successful login attempt was made, but there is no UserID affiliated with the login, send an error message to the administer of the affiliated company (or overarching FieldDay platform's staff) notifying them that either a former employee/user no longer affiliated with the company/platform has attempted login, and to terminate login access.</i>
<i>Pre-conditions</i>	<i>login/signup</i>
<i>Post-conditions</i>	<i>The ID of the current user ID has been retrieved and is associated with the actor.</i>
Name: RetrieveWeekPlanning(CurrentDate)	
<i>Responsibilities</i>	<i>Given the date of login, find out what week number and what corresponding weekly game ID is scheduled for the week. Returns GameID and WeekNumber.</i>
<i>Cross-References</i>	<i>Functional Requirements: UG-S003, UG-S004 Use Cases: Play Game, LeaderBoard</i>
<i>Notes</i>	<i>Input Parameter: CurrentDate requires access to data regarding time zone in order to detect the proper date, as this input will affect the current week number and the active WeeklyGame that will be returned from the WeekPlanning</i>

	<p><i>Database.</i></p> <p><i>Error handling must be included to handle edge cases when a user attempts to login at a cutoff time between two weeks. Preceding week should be returned in this case.</i></p>
<i>Exceptions</i>	<i>If the date does not have a weekly game affiliated with it, return an error message stating that there is no game for the given week.</i>
<i>Output</i>	
<i>Pre-conditions</i>	<i>login/signup</i>
<i>Post-conditions</i>	<i>The current week and corresponding GameID scheduled for the current date has been retrieved from the WeekPlanning and data relevant to the current date has been returned.</i>
Name: RetrieveGame(GameID)	
<i>Responsibilities</i>	<i>Return the ID number of the current user based on their login credentials</i>
<i>Cross-References</i>	<p><i>Functional Requirements: UG-S001, UG-S003, UG-S004</i></p> <p><i>Use Cases: Play Game</i></p>
<i>Notes</i>	<i>Error handling must be included to handle edge cases when a user attempts to start a GameSession at a cutoff time between two weeks. GameID corresponding to the preceding week should be favored in this case.</i>
<i>Exceptions</i>	<i>If there is no GameID that corresponds with the given WeekNumber, indicate that there is no Game for the present week to the user.</i>
<i>Output</i>	<p><i>In the case that there is no WeeklyGame for the current week send a message to administrator prompting a decision on the two options to proceed:</i></p> <p><i>ErrorHandle1: Display message to user: There is no Game for the current week, please check back next week.</i></p> <p><i>ErrorHandle2: allow users to play last active WeeklyGame.</i></p>

<i>Pre-conditions</i>	<i>login/signup</i>
<i>Post-conditions</i>	<i>A new GameSession has been created. The GameSession has been associated with GameID corresponding to the CurrentWeek.</i>
<i>Name: SaveGameSession(GameID, UserID, SesionInformation)</i>	
<i>Responsibilities</i>	<i>Saves the details of the game that was just played in the GameSession History. Saves the GameID that was played, the UserID of who played the game, and the SessionInformation. When all of this input information is successfully stored and saved in the database, a confirmation of successful save is sent.</i>
<i>Cross-References</i>	<i>Functional Requirements: UG-S001, UG-S003, UG-S004 Use Cases: Play Game</i>
<i>Notes</i>	<p><i>This keeps record of all prevalent game details for the sake of having all history for every game played by any user in any company in the GameSession database of the FieldDayPlatform.</i></p> <p><i>SessionInformation includes all data regarding the game: start time, end time, duration, score, and any other information deemed relevant in further iterations in the development of the platform).</i></p> <p><i>ConfirmSave() indicates that all information is successfully stored.</i></p>
<i>Exceptions</i>	<i>If there is an issue with the game being played, information regarding the GameSession attempt was made, along with whatever issue may have occurred (i.e. game termination by user, connection issues, hardware malfunction such as battery loss, etc.) that may have caused the game to quit before results were made.</i>
<i>Output</i>	<p><i>A local and temporary copy of the GameSession is sent to the administer in the case that an error occurs and not all information is saved properly if this information is retrievable.</i></p> <p><i>If the error in saving gameSession information</i></p>

	<i>results in lost information, an error message to the administer will be sent notifying them of the failed attempt to record the gameSession, along with the userID so that the appropriate user can be notified to replay the game.</i>
<i>Pre-conditions</i>	<i>login/signup</i>
<i>Post-conditions</i>	<i>A new Record has been created. The Record is associated with SessionInformation, GameID, and UserID.</i>
<i>Use Case: Leaderboard</i>	
<i>Name: RetrieveUserCompany(CurrentUserID)</i>	
<i>Responsibilities</i>	<i>Returns the company that the current user is working for based off of their login credentials.</i>
<i>Cross-References</i>	<i>Functional Requirements: UG-S001, UG-S002, UF-S003 Use Cases: Display Leaderboard</i>
<i>Notes</i>	<i>An employee or user can not be working at multiple companies simultaneously (i.e can not work for more than one company at a time).</i>
<i>Exceptions</i>	<i>If the user is not currently employed at a company (fired, left, etc.) then indicate that the user is currently unemployed.</i>
<i>Output</i>	<i>If a user is signed in and is employed with a company that utilizes FieldDay, then a success message should be relayed internally within the program. Otherwise, a message should be relayed internally that the current user is not working for a company.</i>
<i>Pre-conditions</i>	<i>login/signup</i>
<i>Post-conditions</i>	<i>The company and their credentials that the user works for has been retrieved and is associated with the actor.</i>
<i>Name: RetrieveGameSessions(Company, GameID, WeekNumber)</i>	
<i>Responsibilities</i>	<i>Returns all of the game sessions (scores, time elapsed, week number, etc.) for all of the employees associated with a certain company.</i>

<i>Cross-References</i>	<i>Functional Requirements: UG-S002, UG-S001 Use Cases: Display Leaderboard</i>
<i>Notes</i>	<i>Every game session from all employees, no matter the level of the employee, will be returned. This means that not only intern or manager game sessions will be returned, but all game sessions.</i>
<i>Exceptions</i>	<i>If no employee's played any games for that particular week due to a holiday for instance, then the returned output should be empty. Additionally, if the actor is currently unemployed, then the returned output will also be empty.</i>
<i>Output</i>	<i>Given that employees of the company have played games for that week and the company does exist (otherwise an empty set will be returned), then a list containing all of the game sessions for the employees of that company for that week will be outputted internally in the system.</i>
<i>Pre-conditions</i>	<i>login/signup</i>
<i>Post-conditions</i>	<i>All game sessions of the employees for the company that the user works for has been retrieved.</i>
<i>Name: ComputeLeaderboard(ListOfGameSessionInstances)</i>	
<i>Responsibilities</i>	<i>Display all of the game sessions for that week for the company that the associated user works for.</i>
<i>Cross-References</i>	<i>Functional Requirements: UG-S004, UG-S002 Use Cases: Display Leaderboard</i>
<i>Notes</i>	<i>Game sessions and scores will only be displayed if employees for that company played games for that week, otherwise the display will be completely empty.</i>
<i>Exceptions</i>	<i>If the user is not currently working for a valid company, indicate that the leaderboard will be empty until the user works for a valid company.</i>
<i>Output</i>	<i>Non-Applicable</i>
<i>Pre-conditions</i>	<i>login/signup</i>

<i>Post-conditions</i>	<i>A display containing all of the game sessions for that particular week for the company that the user works for will be displayed.</i>
Use Case: Employee Metrics	
Name: SetFilters(UserID, StartDate, EndDate, level)	
<i>Responsibilities</i>	<i>Set filters for game sessions and employees based off of various criteria.</i>
<i>Cross-References</i>	<i>Functional-Requirements: UG-S001 Use Case: Employee Metrics</i>
<i>Notes</i>	<i>Non-applicable</i>
<i>Exceptions</i>	<i>Non-applicable</i>
<i>Output</i>	<i>Non-Applicable</i>
<i>Pre-conditions</i>	<i>User for the current session is a manager or has been given privileged access to this information.</i>
<i>Post-conditions</i>	<i>Filters are set for retrieving information.</i>
Name: RetrieveUserGameSessions(UserID, StartDate, EndDate, level)	
<i>Responsibilities</i>	<i>Retrieve game sessions for a specified user(s) given certain criteria</i>
<i>Cross-References</i>	<i>Functional-Requirements: UG-S001 Use Case: Employee Metrics</i>
<i>Notes</i>	<i>Game sessions can be retrieved for a single user or multiple, all depends on which type of filters are turned on or off.</i>
<i>Exceptions</i>	<i>Certain filters are turned on such that no game sessions are received, the returned list will be empty.</i>
<i>Output</i>	<i>A success message will be returned internally for the system if game sessions are returned, or an error message if no game sessions are returned.</i>
<i>Pre-conditions</i>	<i>User is logged in and is a manager or given special privileges.</i>
<i>Post-conditions</i>	<i>A list of all game sessions meeting filtered criteria is returned.</i>

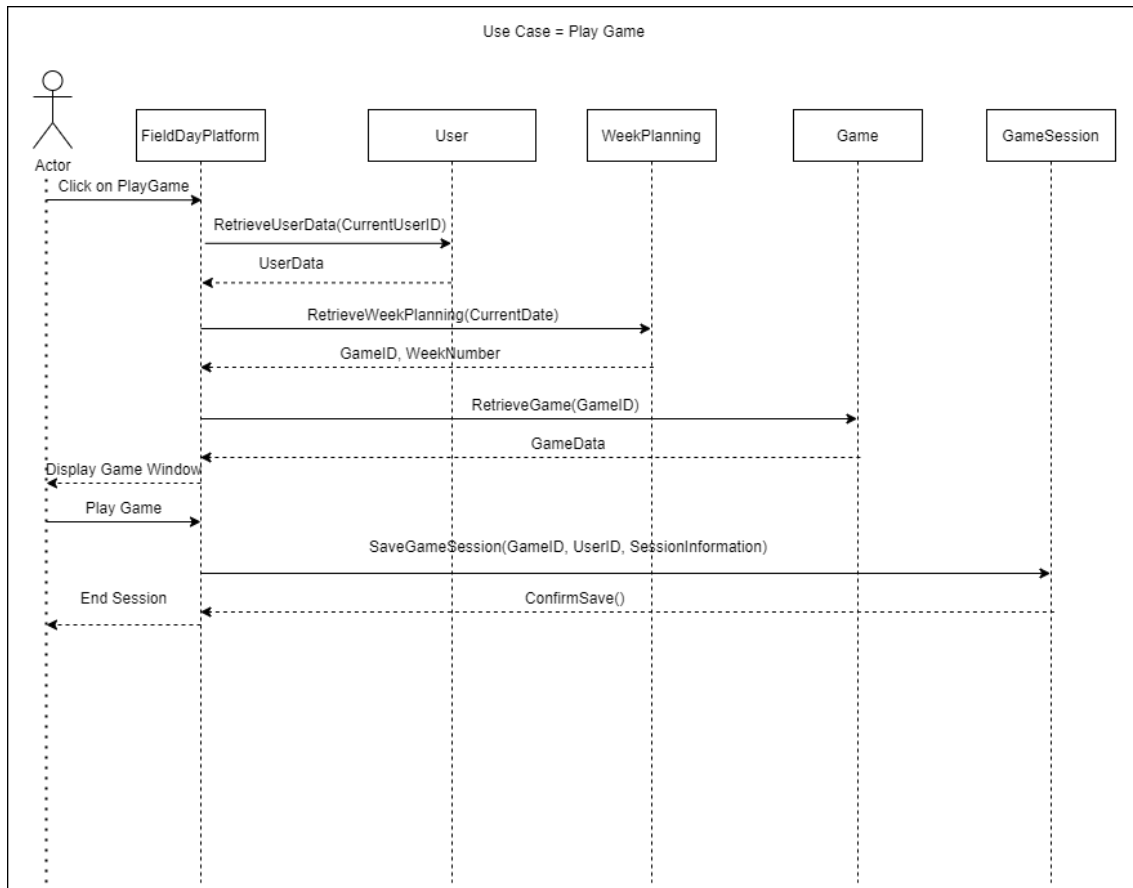
Name: ComputeEmployeeMetrics(ListOfUserGameSessions)	
<i>Responsibilities</i>	<i>Compute trends and statistics for employees based off game session data</i>
<i>Cross-References</i>	<i>Functional-Requirements: UG-S004 Use Case: Employee Metrics</i>
<i>Notes</i>	<i>Employee metrics can only be calculated after a few weeks have already passed.</i>
<i>Exceptions</i>	<i>If the list of game sessions is empty or it's only the second week of games, then there's not enough information to calculate employee metrics.</i>
<i>Output</i>	<i>If employee metrics can be displayed, then a success message will be returned internally within the system (not to the user interface).</i>
<i>Pre-conditions</i>	<i>user is logged in with right credentials, list of user game sessions has been retrieved.</i>
<i>Post-conditions</i>	<i>Analytics in regards to employee satisfaction and participation is acquired.</i>
Use Case: Login/Register	
Name: Login(username, password)	
<i>Responsibilities</i>	<i>Start the login process for the current user, receive the inputted username and password from the user.</i>
<i>Cross-References</i>	<i>Functional Requirements: UG-S005, UG-S001 Use Case: Login</i>
<i>Notes</i>	<i>The hash password needs to be retrieved before the user can be authenticated.</i>
<i>Exceptions</i>	<i>If no information is entered for the username or password input box, then an error will be encountered.</i>
<i>Output</i>	<i>Non-Applicable</i>
<i>Pre-conditions</i>	<i>Input is entered for the username and password</i>
<i>Post-conditions</i>	<i>The hashed password is now able to get retrieved.</i>

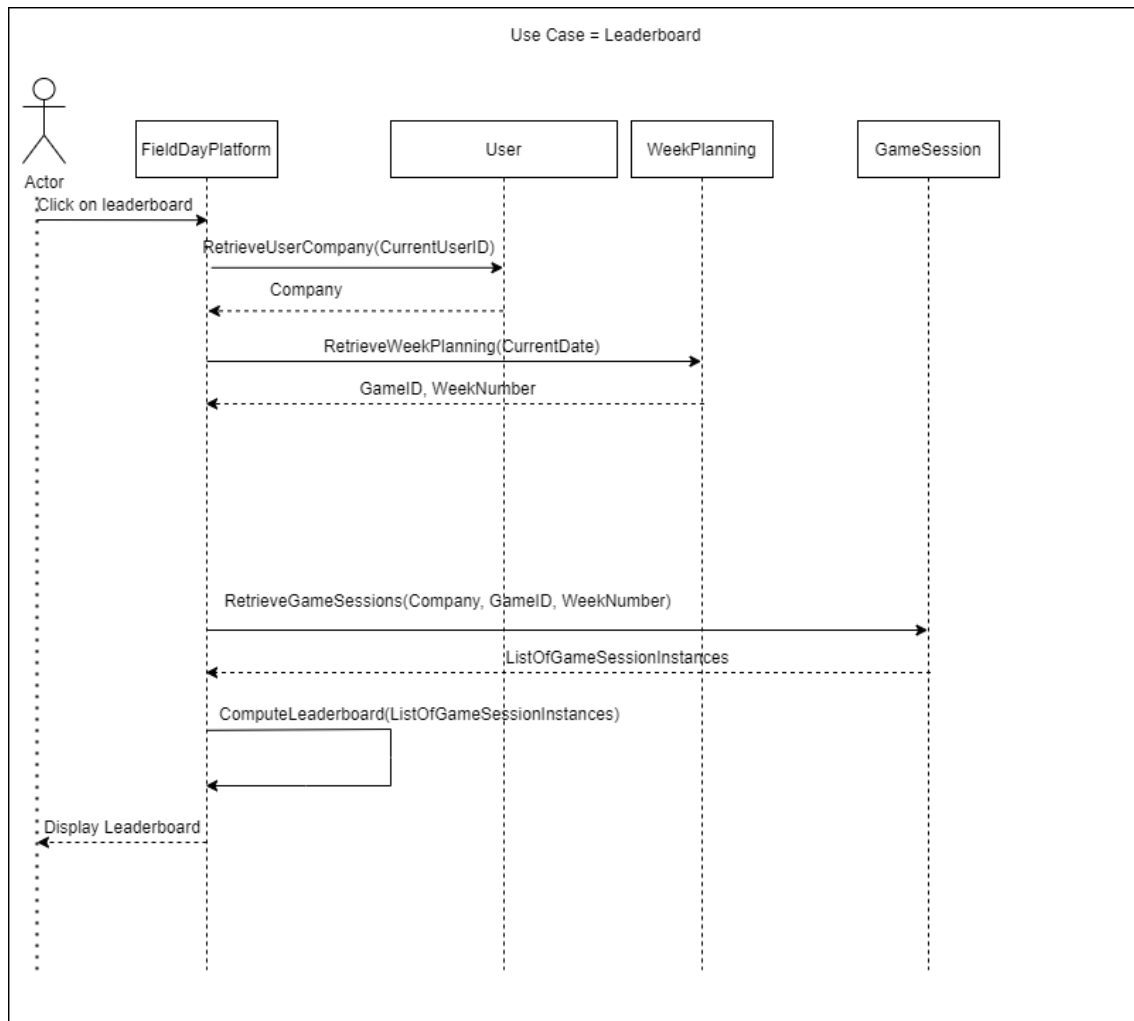
Name: RetrievePasswordHash(username)	
<i>Responsibilities</i>	<i>Return the hashed password from the inputted username from the user.</i>
<i>Cross-References</i>	<i>Functional Requirements: UG-S005, UG-S001 Use Case: Login</i>
<i>Notes</i>	<i>A hashed password will always be able to be retrieved if there is a user with that username. Also, remember that usernames are unique.</i>
<i>Exceptions</i>	<i>If the username inputted does not match a username in the FieldDay database, then an error will be encountered.</i>
<i>Output</i>	<i>If the password hash can not be retrieved, then an error message will be returned internally within the system with the corresponding error.</i>
<i>Pre-conditions</i>	<i>User has inputted a username and password</i>
<i>Post-conditions</i>	<i>The inputted password is now in a hashed format.</i>
Name: Hash(password)	
<i>Responsibilities</i>	<i>Hash the inputted password.</i>
<i>Cross-References</i>	<i>Functional-Requirements:UG-S005, UG-S001 Use Cases: Login</i>
<i>Notes</i>	<i>The retrieved hashed password and the hashed password that's been entered have to match perfectly (case-sensitive) for the login attempt to be successful.</i>
<i>Exceptions</i>	<i>If there is any difference between the two hashed passwords, then the user should not be authenticated.</i>
<i>Output</i>	<i>The password that has been entered by the user is returned in the hashed format.</i>
<i>Pre-conditions</i>	<i>Hashed password has been retrieved</i>
<i>Post-conditions</i>	<i>The inputted password (that's now hashed) is ready to be compared against the retrieved hashed password.</i>

Name: UpdateCurrentUser(username)	
<i>Responsibilities</i>	<i>Make the current session associated with the user that logged in.</i>
<i>Cross-References</i>	<i>Functional Requirements: UG-S005, UG-S001 Use Case: Login</i>
<i>Notes</i>	<i>The current session is only updated with the user if the login credentials are valid. This means that the inputted username and password match an account in the FieldDay accounts database.</i>
<i>Exceptions</i>	<i>If the login credentials are not valid, then the current session should not be updated.</i>
<i>Output</i>	<i>Non-applicable</i>
<i>Pre-conditions</i>	<i>A username and password is inputted</i>
<i>Post-conditions</i>	<i>The current session is updated to user's credentials.</i>

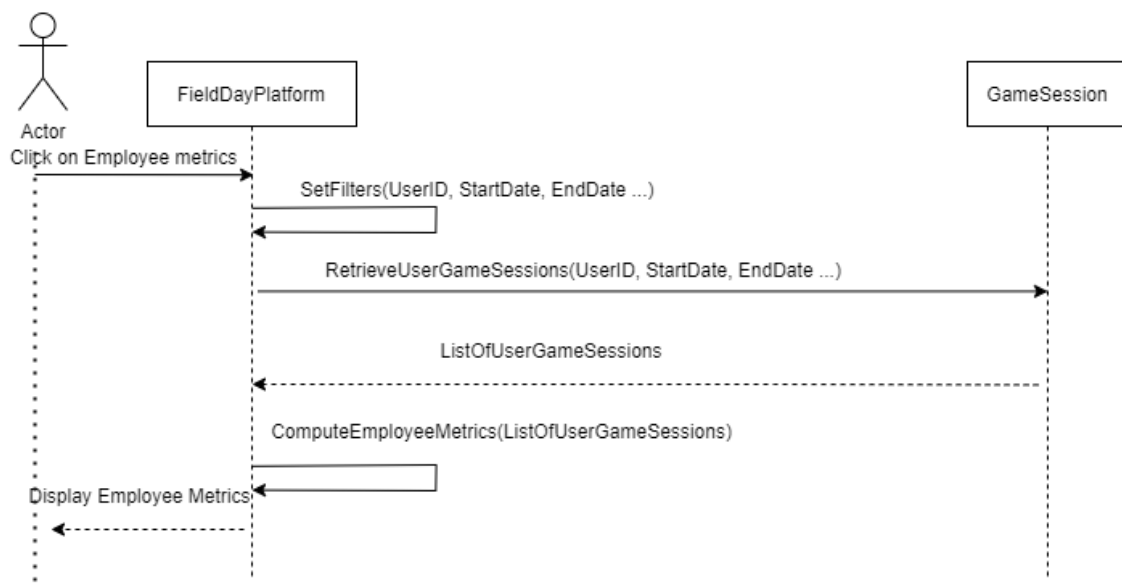
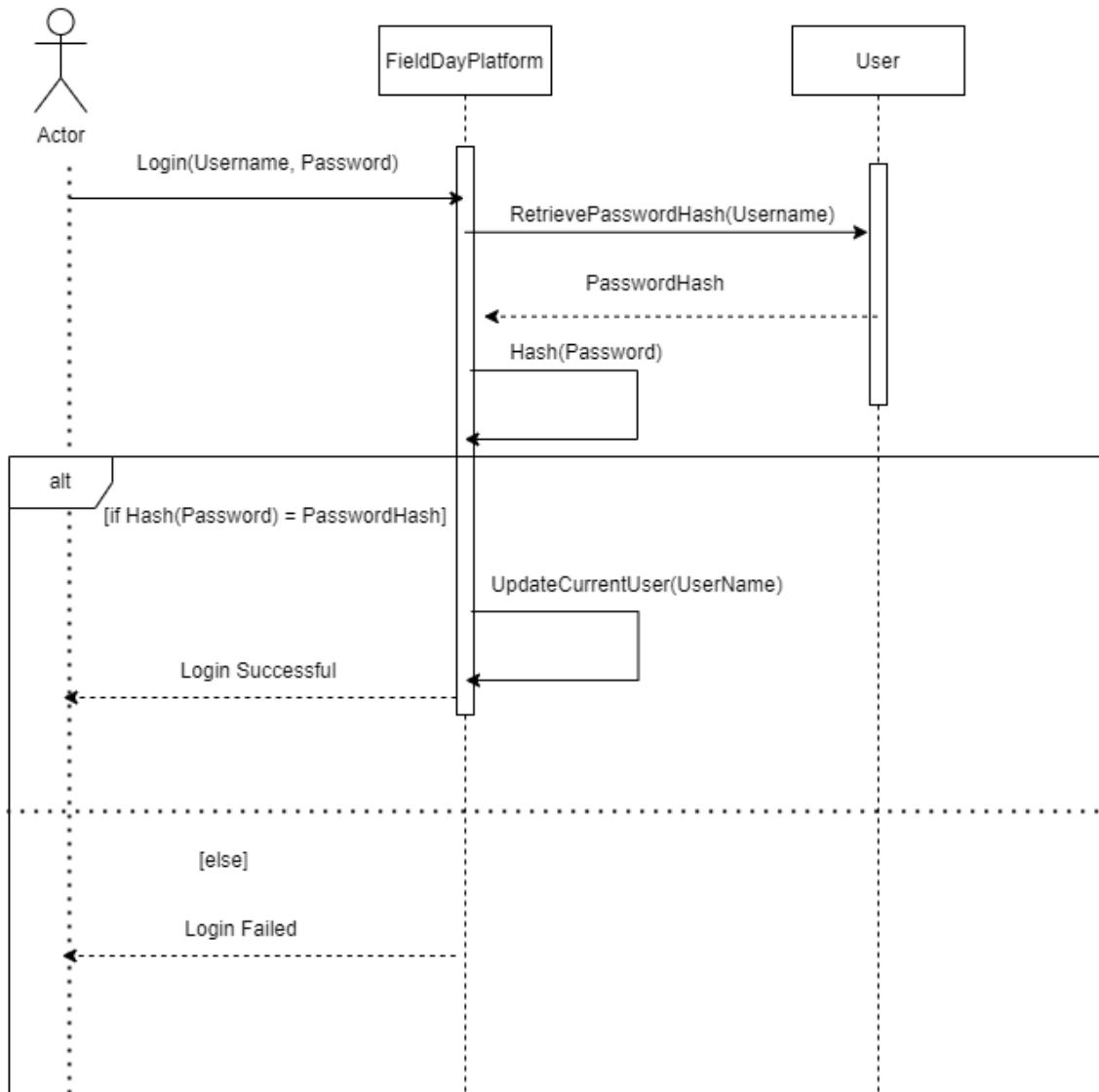
9.1.2 First iteration Design

Sequence diagrams

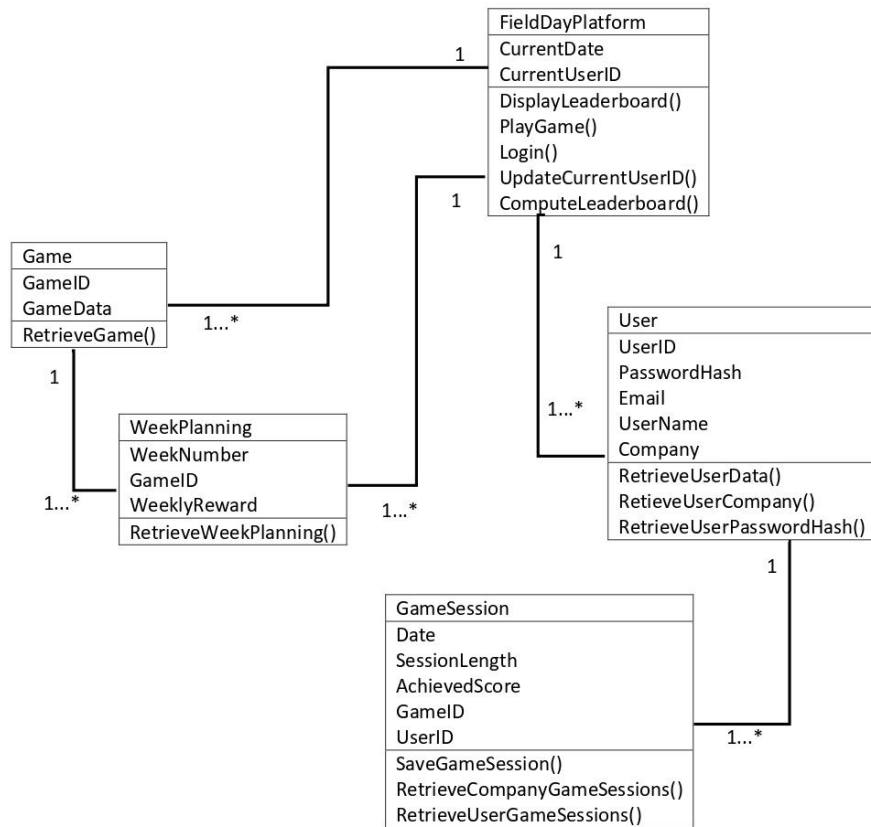




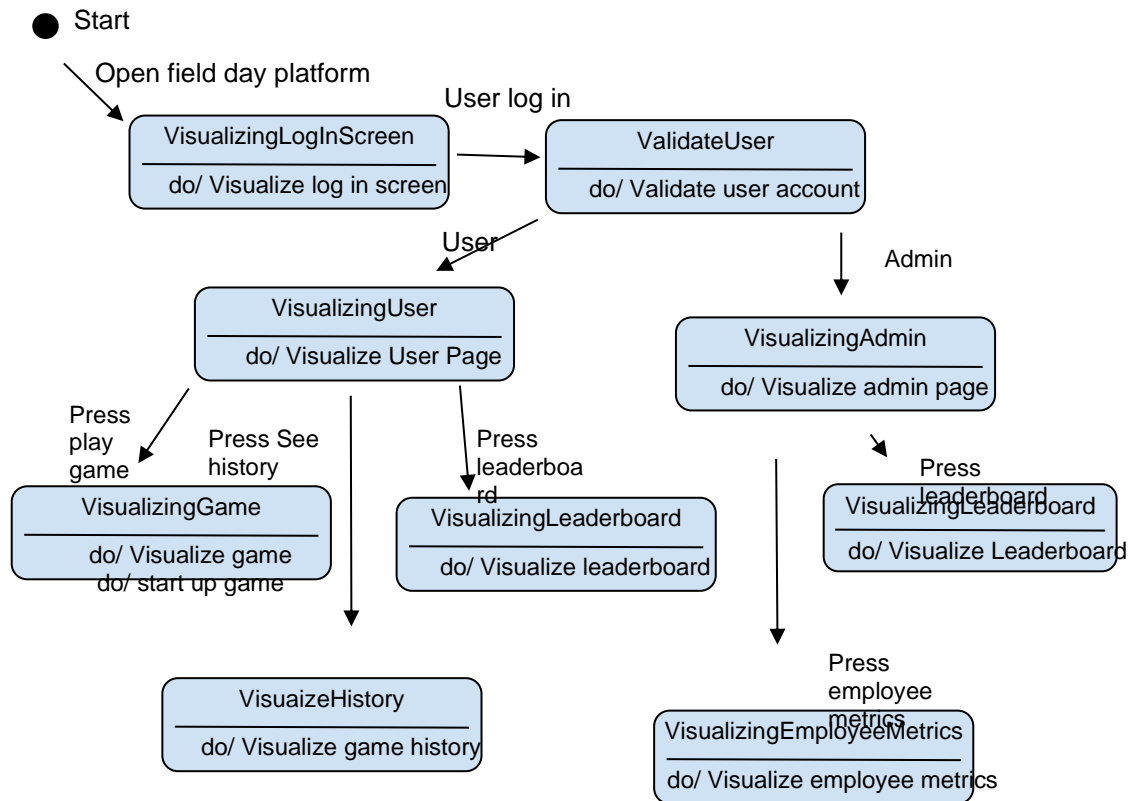
Use case : Login



Class Diagram



Transition State Diagram



9.2 Second Iteration

FieldDay only has one iteration, thus this section is not applicable for our project.

10. Execution of the quality plan

According to the IEEE 730 Quality Plan document, managerial reviews are to be held periodically by management and the CC to assess the execution of all of the actions and the items identified in the SQAP.

The aforementioned management reviews will be performed by the management employees who are directly in charge of the given system or program. For example, a frontend software engineer will carry out the review of a frontend UX interface update request, while a backend software developer is in charge of the managerial review of a database update.

According to the document, this review may sometimes require additional changes in the SQAP itself.

Reviews	Passed/still in process
Review of the Systems Feasibility study	passed

Review of the Use Cases	passed
Review of High Level Use Cases	passed
Configuration Management Plan Review	passed
Review of Project Estimation and Planning <ul style="list-style-type: none"> - estimation - planning 	passed passed
Review of Test Plan	passed
Review of the products of the Analysis Process <ul style="list-style-type: none"> - Use Cases in expanded Format - Review of Conceptual Model Analysis - Review of Operating Contracts 	passed passed passed
Review of Design Process Products <ul style="list-style-type: none"> - Review of Class Diagram - Review of Sequence Diagrams - Review of State Diagrams 	passed passed passed

11. Execution of the configuration management plan

Configuration Elements

CE Code	Name of the CE	CE Description	Date of Creation (Day)	Project to which it belongs	Baseline to which it belongs	Type of CE
0-*-OB	Offer and Budget	The proposed revenues and expenses will be clearly detailed early on and will be evaluated for accuracy and appropriateness throughout the project.	06/04/22	FieldDay	0-4-D	Document

0--QP	Quality Plan	This document clearly details the standards, specifications, requirements, standards to follow, resources allocated, and the priorities for the sequence for the execution of the project.	06/04/22	FieldDay	0-4-D	Document
0--SCMP	SCM Plan	This document outlines the protocol for reviewing, documenting, identifying any requested changes.	06/04/22	FieldDay	0-4-D	Document
0--SCMPR	SCM Plan Review	This outlines the details of the SCM Plan's audits and reviews to ensure that any submitted change requests were performed by the correct personal and fit the requirements as detailed by the plan.	06/04/22	FieldDay	0-4-D	Analysis
0--FUCM	First draft of the use case model	This outlines how the processes will be executed	06/04/22	FieldDay	0-4-D	Model

		and carried out for the project. It defines the major or core elements of the project.				
0-*-E	Estimation	This is the process of predicting the required time, associated costs, and the needed resources for the project.	07/04/22	FieldDay	0-4-D	Document
0-*-ER	Estimation review	This requires analyzing available data to ensure that estimations are accurate and updating them appropriately to ensure fit for the project.	07/04/22	FieldDay	0-4-D	Analysis
0-*-S	Schedule	The plan for the timing, including the start, finish, and duration for each task in a project.	07/04/22	FieldDay	0-4-D	Document
0-*-SR	Schedule review	Reviewing to ensure that the timing of the schedule is appropriate for each task to be successfully	07/04/22	FieldDay	0-4-D	Analysis

		completed in a time and cost efficient manner.				
1-*-FA	Feasibility analysis	This measures and quantifies the success that the project can obtain. It will measure costly success, success with the intended schedule and deliverable date, as well as overall success.	07/04/22	FieldDay	1-4-PR	Document
1-*-FAR	Feasibility review	This review ensures that the project will be able to accomplish the goal of reducing burnout in its users.	08/04/22	FieldDay	1-4-PR	Analysis
0-*-UCM	Use cases model	This outlines the different use cases (in this project there are four) and how each actor interacts with them. Dependencies are detailed using arrows to show the connection between the nodes (aka	06/04/22	FieldDay	1-4-PR	Model

		the use cases)				
0-*-UCMR	Quality Plan Review	Reviews the standards, specifications, requirements, standards to follow, resources allocated, and the priorities for the sequence for the execution of the project.	06/04/22	FieldDay	1-4-PR	Analysis
1-*-PUC	Prioritization of use cases	This identifies the high priority use cases based on the goals of the client and the intended effect they want the project to have on its users.	08/04/22	FieldDay	1-4-PR	Document
1-*-PUCR	Prioritization of use cases review.	This review ensures that the priorities are indeed in alignment with the clients needs and will lead to the success of the project as a whole.	08/04/22	FieldDay	1-4-PR	Analysis
1-*-DHLUC	Definition of high-level use cases.	These are written to properly match the various	08/04/22	FieldDay	1-4-PR	Document

		scenarios of different situations that could occur in the project.				
2-*-UCEF	Use cases in extended format	These describe how the project will operate in situations other than the main success scenario (aka when everything goes as planned). The extended use cases outline how the project will interact in cases when things do not go right or as planned.	11/04/22	FieldDay	1-4-PR	Document
2-*-UCEFR	Use cases in extended format review	This review ensures that all edge cases are accounted for and that none are missed to make sure that in every scenario, the project will be able to successfully execute and operate.	11/04/22	FieldDay	1-4-PR	Analysis
2-*-CM	Conceptual model	This helps describe the relationships	11/04/22	FieldDay	2-4-C	Model

		of the different datasets collected and used in the project.				
2-*-CMR	Conceptual model review	This review checks that the model properly describes the real-life social aspect of the scenarios and process within the project.	12/04/22	FieldDay	2-4-C	Analysis
2-*-OC	Operation Contracts	This details the contract between the owner of the project and how it will be used by the client.	11/04/22	FieldDay	2-4-C	Document
2-*-OCR	Operation Contracts review	This makes sure that all cases are covered by the contract to ensure proper use of the project.	11/04/22	FieldDay	2-4-C	Analysis
2-*-CD	Class diagram	This outlines the different classes of the project.	12/04/22	FieldDay	2-4-C	Document
2-*-CDR	Class diagram review	Ensure that all parts of the project are included including as cases.	12/04/22	FieldDay	2-4-C	Analysis

2-*-SD	Sequence diagrams	This depicts the various activities within the project in a graph to depict their relationships.	12/04/22	FieldDay	2-4-C	Model
2-*-SDR	Sequence diagrams review	Ensure that this is accurate so that a better sense of timing can be obtained.	12/04/22	FieldDay	2-4-C	Analysis
2-*-TSD	Transition states diagram	This details all the different states the project can have, the events/scenarios in which these states can change, and what conditions need to be satisfied before a state can change.	12/04/22	FieldDay	2-4-C	Document
2-*-TSDR	Transition states diagrams review	This ensures that all conditions are met before states are changed and that the state changes coincide with the proper use of the project.	13/04/22	FieldDay	2-4-C	Analysis

2-*-FEC	Front End Coding		13/04/22	FieldDay	2-4-C	Development
2-*-FECR	Front End Coding Review		27/04/22	FieldDay	2-4-C	Development
2-*-BEC	Back End Coding		13/04/22	FieldDay	2-4-C	Development
2-*-BECR	Back End Coding Review		11/05/22	FieldDay	2-4-C	Development

Baseline

Baseline name	Baseline code	Baseline description	Baseline date of creation	Baseline CIs	Baseline Allocation
Design	0-4-D	This baseline sets up the design of the project including the first use cases, budget, estimations and overall plan. The phases following are dependent on the drafts from this baseline.	06/04/22	Offer and Budget Quality Plan SCM Plan SCM Plan Review First draft of the use case model Estimation Estimation review Schedule Schedule review	22 hours
Planning and requirements	1-4-PR	This baseline refines and builds off the plan from the previous baseline. The use case model is expanded on and specificity is added. The original plan is reviewed and broken up by priority of each aspect.	07/04/22	Feasibility analysis Feasibility review Use cases model Use cases model review Prioritization of use cases Prioritization of use cases review. Definition of high-level use cases. Definition of high-level use cases review	17 hours

Construction	2-4-C	This baseline builds on the use case model and adds additional design. And planning aspects to the project.	11/04/22	Use cases in extended format Use cases in extended format review Conceptual model Conceptual model review Operation Contracts Operation Contracts review Class diagram Class diagram review Sequence diagrams Sequence diagrams review Transition states diagram Transition states diagrams review	175 hours

Dependencies

CI-1 Code	CI-2 Code	Date
1-*-UCM	0-*-E	06/04/22
0-*-OB	1-*-FA	06/04/22
1-*-UCM	1-*-PUC	06/04/22

Successions

CE Code	Previous Version	Next Version	Date Finished
0-*-OB	None	1	06/04/22
0-*-QP	None	1	06/04/22
0-*-SCMP	None	1	06/04/22

0-*-SCMPR	None	1	07/04/22
0-*-FUCM	None	1	06/04/22
0-*-E	None	1	07/04/22
0-*-ER	None	1	07/04/22
0-*-S	None	1	07/04/22
0-*-SR	None	1	07/04/22
1-*-FA	None	1	07/04/22
1-*-FAR	None	1	08/04/22
0-*-UCM	None	1	06/04/22
0-*-UCMR	None	1	06/04/22
1-*-PUC	None	1	08/04/22
1-*-PUCR	None	1	08/04/22
1-*-DHLUC	None	1	08/04/22
2-*-UCEF	None	1	11/04/22
2-*-UCEFR	None	1	11/04/22
2-*-CM	None	1	11/04/22
2-*-CMR	None	1	12/04/22
2-*-OC	None	1	12/04/22
2-*-OCR	None	1	12/04/22
2-*-CD	None	1	12/04/22
2-*-CDR	None	1	12/04/22
2-*-SD	None	1	12/04/22
2-*-SDR	None	1	12/04/22
2-*-TSD	None	1	12/04/22
2-*-TSDR	None	1	13/04/22
2-*-FEC	None	1	13/04/22
2-*-FECR	None	1	27/04/22

2-*-BEC	None	1	13/04/22
2-*-BECR	None	1	11/05/22

Derivations

CI-1 Code	CI-2 Code	Date
1-*-UCM	0-*-E	06/04/22
0-*-FUCM	0-*-OB	06/04/22
1-*-FA	1-*-UCM	08/04/22
1-*-UCM	1-*-PUC	06/04/22
1-*-UCM	1-*-DHLUC	06/04/22