Software Development Project Management

Date: 29/04/2022 Group: 1
Karen Dahl Aarhus
Emilie Lia-Rognli
Sigrun Asheim Nummedal
Julie Holte Motland

Client: Maribel Inc.

Content

Tables index

Figures index

CREATIVE IDEA DESCRIPTION

The idea is a Virtual Personal Assistant (VPA) designed to both prevent burnout in its users by enforcing healthy habits as well as picking up on the early signs of burnout by analyzing the user's interaction with the device. The VPA will be called Charlie, a gender neutral name and will by default also have a gender neutral voice.

Common causes of burnout is lack of social support, taking on more than one can handle and poor self care. A stressful and emotionally tiring working environment is also often linked to burnout, as well as long and most importantly continuous working hours. If employees receive emotional support from colleagues, take breaks and not take on more tasks than they can handle, many cases of burnout will be prevented.

The product will be targeted towards companies with employees working from home and therefore are more prone to lose track of time, forget to take breaks and socially interact with their colleagues than their peers who work in an office environment surrounded by people. Charlie will sync with the user’s calendar to fit into its workday as seamlessly as possible to not be a distraction or annoyance. This includes muting messages during meetings and suggesting breaks at times when it is possible for the user to carry out. Charlie will over time collect data about the users responses and reactions and tailor its recommendations and notifications to the users needs and feelings.

The key aspect of the VPA is to provide the users with someone who helps them handle the situations that put them at risk for burnout symptoms. Charlie will help and teach the user to set boundaries, show understanding and tailor their notifications, reminders and tips towards the user’s needs. Charlie will keep track of the user's mental health on a score of 1-10. Through weekly check-ins, simple conversations, analysis of habits and responses to questions throughout the workday, Charlie will build a score of how it perceives the user's mental state. The content provided will be based on this score, so if a user's score is nearing a level indicating symptoms of burnout it will receive more reminders about breaks, meditation, breathing exercises, home stretching etc. If a user hits level 2 or 3 Charlie will recommend talking to a manager or therapist to further work on the stress that it is experiencing.

The product can be personalized by the user. The user can choose to change the voice of Charlie to either a female or male voice. The user will also be able to customize the intervals of messages and reminders from Charlie, as well as change the fixed time set for the weekly check-in.

In order for Charlie to be helpful and seen as a positive addition to the user's workday it has to be offered and optional, i.e. not forced. The idea is for the companies to offer the product to their employees as a perk they can choose to

take advantage of, just like many workplaces nowadays offer other bonuses in order to keep their employees happy and healthy. Examples are workouts during office hours, free fruit and subsidized lunch, which are all now normal things to encounter in a workplace, and all put there intended to increase the health, happiness and productivity of its workers. Charlie will act as an additional perk to help workers maintain a balanced worklife where their mental health is prioritized.

1. General data of the company offering the project

Name: Charlie Inc.

Description: Charlie Inc. will assist in the work life of an employee working from home. The company's main product, the VPA, especially aims to prevent burnout by picking up early signs in order to catch it before it is too late. This device is useful for every company that has employees working from home that want to prevent burnout and have a low sick leave.

Mission: The main mission of the company is to prevent burnout as this is an increasing problem in the working life. The burnout syndrome should be taken seriously and prevented before it is being developed. The mission of the VPA is to early detect symptoms of burnout syndrome and help and assist in the everyday work life of an employee. The goal is to both prevent burnout by picking up early signs in order to catch it before it is too late, and also assist the users in its working life.

2. Definitions and acronyms

VPA: Virtual Personal Assistant
Charlie: The VPA's name

3. Initial offer and budget

3.1 Offer

Charlie Inc. offers a VPA to Maribel AS to prevent their employees from getting burnouts. Mainly, the VPA will work as an assistant to catch early signs of burnout and prevent it by making sure employees take breaks and have a good work routine. The first goal is to have a working product where the only communication with the assistant is through talking. If the client is satisfied with the first product we will implement features making the product usable for people with hearing or talking disabilities.

There are multiple other voice assisted products on the market, but none that we know of that shares the goal of Charlie. Products such as Alexa and Google Home both aim to assist its users with easy tasks in everyday life, but do not offer any assistance in taking care of the users mental health or promoting healthy habits. There are also multiple applications on the market either made to guide you through a meditation or remind you to drink water or take a break, but none of these are tailored to the users specific needs and schedules. That is where Charlie

stands out and becomes the best. By using AI technology the VPA will analyze user behavior, tailor its preventative measures to the person's needs and its schedule.

Based on the use cases, the project lifecycle will last for 6 months including testing and deployment. To get the best result we will have a multidisciplinary team working on the product. The team will consist of psychologists, designers, software engineers, testers and developers. Psychologists will only be used at the start of the project, then the rest of the team resources will be adjusted based on the main task of the project. The central part of the product development will be done by the end of month 4. The remaining months will be used to make adjustments, test and deploy to make sure the product is of good quality.

3.2 Budget

3.2.1 Use case model

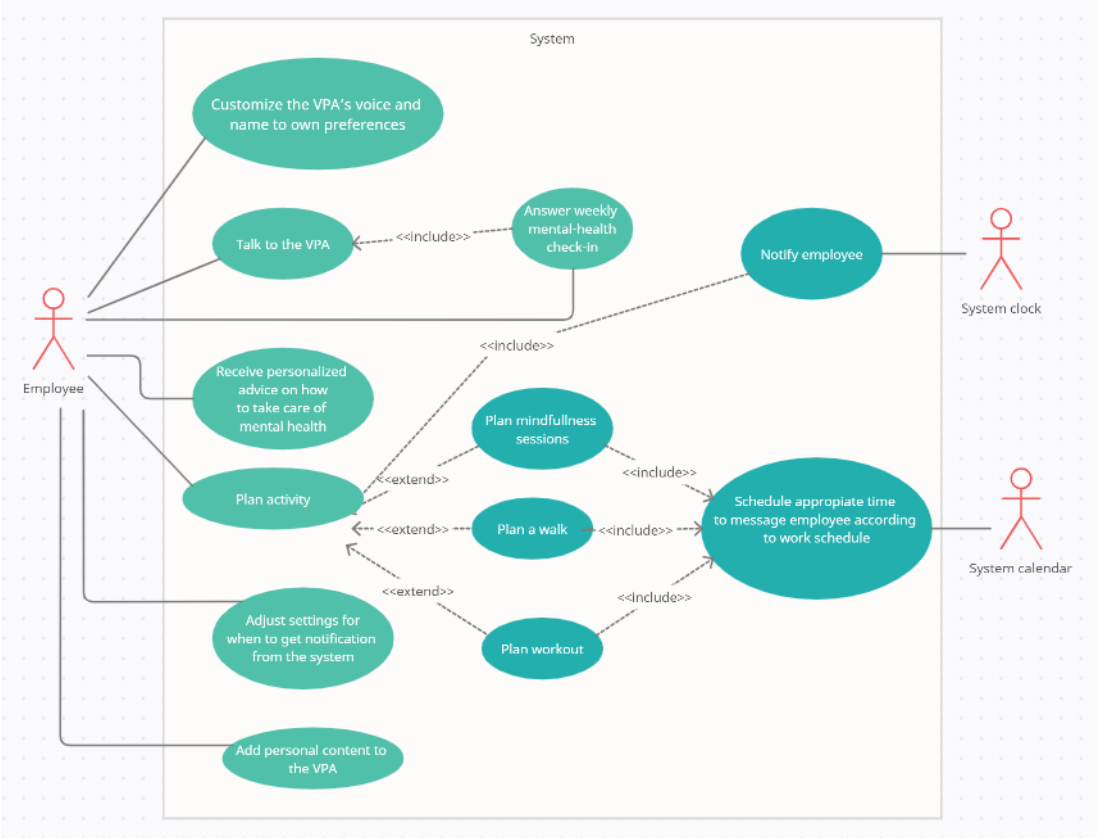


Figure 1: Use case model

3.2.2 Use case estimation

To estimate the time the project will take to develop we use the use cases. As we do not have any company history or experts at hand, we will use estimation techniques and adjust based on the complexity of each use case. The following table (Table 1) contains all the use cases and their corresponding estimated complexities.

Table 1: Use case complexities

Complexity	Use case
Complex	Talk to the VPA
Average	Receive personalized advice on how to take care of mental health
Average	Plan activity
Simple	Adjust settings for when to get notifications from the system
Simple	Customize the VPA's voice and name to own preferences
Average	Add personalized content to the VPA
Simple	Answer weekly mental health check-in
Average	Notify employee
Complex	Schedule appropriate time to message employees according to work schedule
Average	Plan walk
Simple	Plan workout
Simple	Plan mindfulness sessions

Based on the estimated complexities the development duration is estimated as follows in Table 2. The rightmost column shows the calculated adjusted time for each type of use case. By summing the months we get a total development duration of 4,125 months.

Table 2: Estimation of use case duration

Use case type	Description	Number of cases	Adjustment factor	Time (months)
Simple	1-3 transaction	5	1	1,25
Average	4-7 transactions	5	1,5	1,875
Complex	> 8 transactions	2	2	1

3.2.3 Salary of the team

We decided the composition of the team by analyzing the use cases in Table 1 and our estimations. The first month, which will be the research and prototyping phase, the team will consist of 1 designer, 1 psychologist, 1 software engineer, and 1 developer. For the development phase of the project, we estimated that we will need 1 designer, 2 software engineers, and 4 developers. This phase will last for approximately 4 months, calculated in section 3.2.2. The testing of the solution will overlap with the development phase and we estimate that the team will only need 1 software tester. For the deployment of the solution, only 1 software engineer will be needed.

The monthly salaries are based on average salaries for the given professions in Spain.

Table 3: Salary estimation

Occupation	Monthly salary	Research & prototyping	Development (4 months)	Testing	Deployment	Total salary
Designer	800€	1	1			4.000€
Psychologist	2.000€	1				2.000€
SE	1.200€	1	2		1	12.000€
Developer	900€	1	4			15.300€
Software tester	900€			1		900€
					Total sum	**34.200,00 €**

3.2.4 Computer equipment and software

There will be a maximum of 7 people working with a computer at the same time, which can be seen by looking at the column “Development” in Table 3. Therefore the company will need to have 7 sets of computers and equipment. The price for a Dell XPS 13 without IVA is 1.350€, giving a total of 9.450€ for 7 computers. In addition equipment such as a keyboard and mouse is needed, which can be bought as a combo from Logitech for 18€, or 126€ for 7 sets. In total, this adds up to 9.576€ in expenses.

With regards to software expenses, the computers will come with built-in operating systems. In addition, developers and software engineers can use free software during development, such as free editors and GitHub to share code. For the designing process Figma, which is a free and well-known software, can be used. Hence, the software expenses are 0€.

3.2.5 Summarized budget

In the following table (Table 4), the summarized budget is presented. In addition to the previously mentioned posts, some more are present in the budget. For consumables, the team has 3.000€ to buy what they need for office equipment. The travel expenses are for the end-user testing during the testing phase, both for the end-users that will travel to the team office and for when the software tester needs to travel and book a space for testing near the end-users.

Table 4: Budget for the project

Description	Total

Salary of the teamwork	34.200,00 €
Computer equipment	9.576,00 €
Consumables	3.000,00 €
Travel and Expenses	1.000,00 €
Software	0,00 €
VAT (21%)	2.850,96 €
Risk (15%)	7.166,40 €
Profit (10%)	4.777,60 €
Total	62.570,96 €

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 Charlie the VPA.

4.3 Definitions and Acronyms

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

VPA: Virtual Personal Assistant
Charlie: The VPA's name
QM: Quality Manager
QR: Quality Responsible

4.4 References

This plan is based on the description of SCM planning that can be found in the IEEE 828-2012 standard.

SMANAGEMENT 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 section 4.8.5.

4.6 Responsibilities

Change control committee: *Ingrid Amalie Solbjørg and Solveig Hergot Langås*

Responsible for SCM: *Julie Holte Motland and Karen Dahl Aarhus*

Librarian: *Sibora Baboci and Emilie Lia-Rognli*

Rest of the development staff: *Sigrun Nummedal*

4.7 Applicable policies, directives and procedures

The applicable procedures are described in the section: "4.9 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

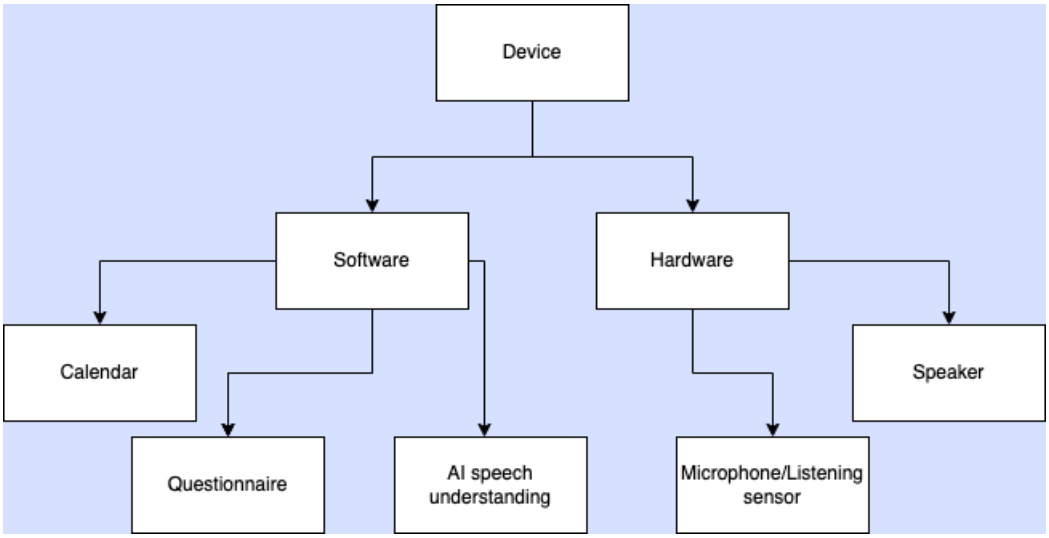


Figure 2: General system structure

4.8.2 Selection of the configuration elements

The configuration elements (CE) will be the output from the Larman methodology:

- Specification of the system
- Software project plan
- Prototype
- Preliminary design
- Detailed design
- Executable programs
- User's manual
- Operation and installation manual
- Test cases & results recorded
- Software engineering standards and procedures used
- Problem reports
- Maintenance requests
- Hardware and software products used during development
- Documentation and manuals of the hardware and software products used during development
- Database designs
- Database contents
- Use case model
- Class model
- Sequence diagram
- Offer
- Budget
- SCM plan
- Quality plan
- Software code

4.8.3 Selection of the identification scheme

Table 5 displays how the identification scheme will look like.

Table 5: Identification scheme

CE code	CE	Description	Creation date	Project	Baseline	Type of CE

4.8.4 Definition of relationships

Succession
Date is when it is created.

Table 6: Succession

CE code	Previous version	Next version	Date

Dependency
CE code 1 is dependent on CE code 2. Date is when it is created.

Table 7: Dependency

CE code 1	CE code 2	Date

Derivation
CE code 1 is derived from CE code 2. Date is when it is created.

Table 8: Derivation

CE code 1	CE code 2	Date

4.8.5 Definition and establishment of baselines

Table 9: Baselines

Baseline name	**Description**
Analysis	Collecting necessary information to create the requirements and use cases. As well as talking to experts to get a useful product. This involves many plans and models, as well as budget and offer.
Design	Detailed design documents (ex. figma-sketches).
Implementation	Delivery will include executable code units, user and maintenance manuals, test procedures, test cases and test reports.
Iteration 1	Completion of what has been done in iteration 1. When iteration 1 is done the coding, testing, analysis and design related to iteration 1 should be delivered. For example some code is delivered and a plan for the next iteration is made.
Iteration n	Completion of what should be done in iteration n. Delivery same as in iteration one.

4.8.6 Definition and establishment of software libraries

server/charlie/work/:employee_id - personal workspace for team members to keep notes, documents, etc.

server/charlie/integration - to integrate CEs.

server/charlie/support - in this folder you will find the finished CEs.

server/charlie/code/dev - in this folder all the code will be located during development before it is deployed to server/charlie/code/prod.

server/charlie/code/prod - in this folder all the code for the production (that is deployed to the user) will be located. Is also the client version of the product.

4.9 Changes control

Change Control Procedure:

1. Initiation of change: the request for change, duly completed by the applicant, shall be submitted.
2. Classification and registration of the request for change.
3. Evaluation and Approval or rejection by the Change Control Committee.
4. In case of approval, notification to the originator and to the managers of the CEs concerned.
5. The change is made by entering a monitoring and control process.
6. Once the change has been made, the change control committee certifies that it has been made correctly.
7. Finally, the originator of the change is notified of this certification.

Change Request Report Format

<table>
<tr><td colspan="3">Change Request Report</td></tr>
<tr><td colspan="2">System name:
System code:</td><td>Level of implementation of change:
System:
Hardware:
Software:
Documentation:
Another one:</td></tr>
<tr><td>Name of the applicant:

Phone:

Date of application:</td><td>Priority of change:
Routine:
Urgent:
Very urgent:</td><td>Are other hardware or software systems affected?
YES
NO</td></tr>
</table>

Description of the change: **Need for change:** **Estimation of the effect of the change on other systems, software and equipment** **Alternatives to change:**	
To be filled in by the change control team	
Date of receipt of application:	**Disposition:**
Signed:	**Date:**

Report 1: Change Request Report

Change Certification Report Format

Certification of the change:
Date of certification: **Originator:** **Recipient:**
Results obtained **Signed:**

Report 2: Change certification report

4.10 Status account

For the identification scheme we will use a significant identifier for each CE. The following structure for the CE code will be used:

Xn-YY, where XX is the phase identifier, n is the iteration number where 0 is the first initial planning phase in the Larman method, and YY is the acronym for the CE.

X can be:

A - Analysis phase
I - Implementation phase
T - Testing phase

Table 10: Status account

CE code	CE	Description	Creation date	Project	Baseline	Type of CE
A0-UCM	Use case model	A model with the different things an actor can do with our system. In total there are 13 use cases.	16.02.22	Charlie	Analysis	Model
A0-SS	System Specification	System requirements based on the use case model.	14.03.22	Charlie	Analysis	Document
A0-CM	Class Model	Class Model based on the use cases.	29.04.22	Charlie	Analysis	Model
A0-SD	Sequence Diagram	Sequence diagram based on the use cases.	29.04.22	Charlie	Analysis	Model
A0-OF	Offer	Offering a virtual personal assistant to help prevent burnout in the workplace.	20.02.22	Charlie	Analysis	Document
A1-BU	Budget	The budget was created based on the estimation of the use cases and their complexities. Then the salary of the team was estimated to be 34.200€. By adding indirect costs and for computer equipment the total budget sums	20.02.22	Charlie	Analysis	Document

		up to 62.570,96€.				
A0-SCM	SCM Plan	The SCM plan tells how the activities should be carried out and checks if the quality has been reached.	01.03.22	Charlie	Analysis	Plan
A0-QP	Quality Plan	Quality plan is saying how to review and what is expected from each process.	01.03.22	Charlie	Analysis	Plan
A0-SPP	Software project plan	The software project plan includes the gantt chart made of the whole project	08.04.22	Charlie	Analysis	Plan

Succession
Date is when it is created.

Table 11: Succession

CE code	Previous version	Next version	Date
A1-BU	A0-BU	A1-BU	01.03.22

Dependency
CE code 1 is dependent on CE code 2. Date is when it is created.

Table 12: Dependency

CE code 1	CE code 2	Date
A0-CM	A0-UCM	29.04
A0-SD	A0-UCM	29.04
A0-CM	A0-SD	29.04

Derivation
CE code 1 is derived from CE code 2. Date is when it is created.

Table 13: Derivation

CE code 1	CE code 2	Date
A0-OF	A0-UCM	20.02.22
A1-BU	A0-UCM	20.02.22
A0-SS	A0-UCM	14.03.22

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)

- Ingrid (QM) did a review of the feasibility study on the 22nd of March
- Ingrid (QM) did a review of the configuration management plan on the 22nd of March
- Julie (QR) did a review of the use cases in the expanded format on the 1st of April
- Julie (QR) did a revision of the use case diagram on the 26th of April
- Ingrid (QM) did a review of the high-level use cases on the 26th of April
- Julie (QR) did a revision of the estimation on the 26th of April
- Ingrid (QM) did a revision of the planning on the 26th of April
- Ingrid (QM) did a review of the operating contracts on the 29th of April
- Julie (QR) did a revision of the class diagram on the 29th of April
- Ingrid (QM) did a revision of the sequence diagram on the 29th of April
- Ingrid (QM) did a revision of the state diagrams on the 29th of April

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 Ingrid Amalie Solbjørg as Quality Manager and Sibora Baboci 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

Ingrid Amalie Solbjørg, 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.

Julie Holte Motland, as Quality Responsible, must carry out the revision of the Use Case Diagram, for this she 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

Ingrid Amalie Solbjørg, 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

Ingrid Amalie Solbjørg, 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.

Julie Holte Motland, 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.
Ingrid Amalie Solbjørg, 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

Ingrid Amalie Solbjørg, 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

Julie Holte Motland, 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

Ingrid Amalie Solbjørg, 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

Ingrid Amalie Solbjørg, 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.

Julie Holte Motland, 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

Ingrid Amalie Solbjørg, 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

Ingrid Amalie Solbjørg, 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

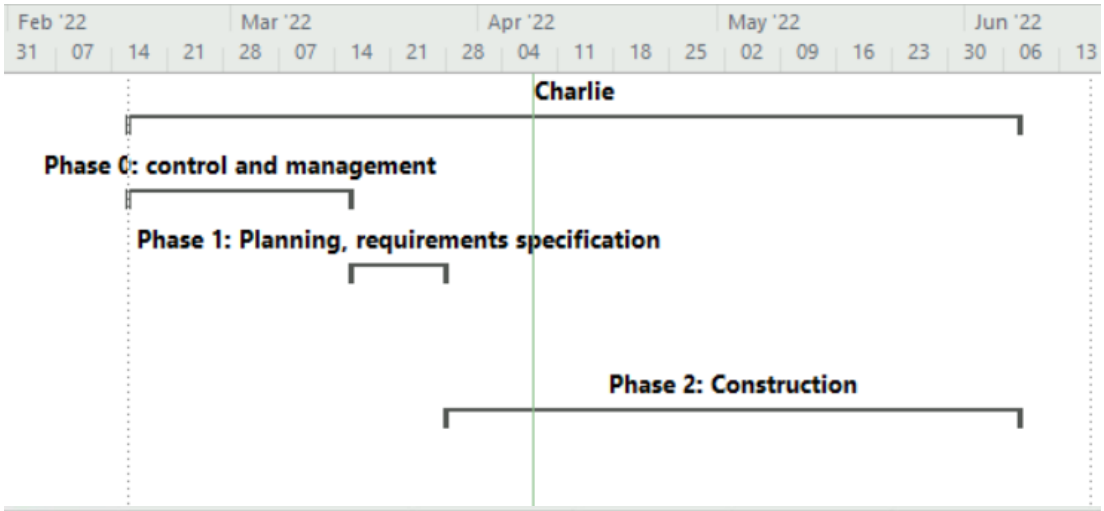
The estimation can be seen in detail with the provided excel sheet.

Table 14: Estimation

Adjusted Use Case Points (UCP) = UUCP * TCF * EF				85.4865	
Person Hours Multiplier (PHM) (Per use case)	* A value of 0 means too risky to proceed			20	hours.use-case
Effort in Person Hours = UCP * PHM (just coding)				1709.73	hours.man
Effort in Person Hours whole project				4274.325	hours.man
Esfuerzo Meses Persona Estimados en el Proyecto				36.84762931	MM
Time estimated using COCOMO II Organic Mode	Tdev=2.5(MM)0.38			9.844091421	Months
Average Team Size (Full Time)	Team Size =MM/Tdev			3.743121405	People
Cost				37989.90582	Euros
Hours worked per month	116				
Average Monthly Salary (euros)	1031				

7. Planning

The Gantt chart represents all the tasks that will be done in the different phases of the project. An overview of the project's phases is presented below:



Task Mode	Task Name	Duration	Start	Finish	Predecessors
	Charlie	80 days	Wed 16/02/22	Wed 08/06/22	
	Phase 0: control and management	20 days	Wed 16/02/22	Wed 16/03/22	
	Phase 1: Planning, requirements specification	8 days	Wed 16/03/22	Mon 28/03/22	15
	Phase 2: Construction	52 days	Mon 28/03/22	Wed 08/06/22	23

Figure 3: Extraction of Gantt chart

The Gantt chart can be seen in detail with the provided Microsoft project.

8. Planning and requirements specification

8.1 Feasibility study

IDENTIFYING THE SCOPE OF THE SYSTEM

Charlie will be a Virtual Personal Assistant (VPA) to help prevent burnout for people working from home. The VPA will keep track of a user's mental health by weekly check-ins and using this data to give personalized notifications and reminders to fit the user's needs. It will also sync with the user calendar so that it can customize settings after the work day, such as no distractions or notifications during meetings. Charlie will be sold through companies that want to offer the solution to their employees.

IDENTIFICATION OF STAKEHOLDERS IN THE SYSTEM

A stakeholder is someone with interests in a system. The stakeholders consist of two companies; the client Maribel Inc. and the company which offers the project, Charlie Inc. In addition the users of the systems are its stakeholders, and their employers.

Also, the following people, who are going to carry out the project, are stakeholders:

- **Project Manager:**
  - Sibora Baboci
- **Analyst:**
  - Karen Dahl Aarhus
- **Designer:**
  - Emilie Lia-Rognli
- **Programmer:**
  - Sigrun Nummedal
- **Tester:**
  - Solveig Hergot Langås
- **Configuration Management:**
  - Julie Holte Motland
- **Quality Management:**
  - Ingrid Amalie Solbjørg

8.1.1 Requirements definition

The requirements are going to be described as follows:

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 nnn indicates the number
  - 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: G-C001	
Name:	The system shall be capable of audio input.
Priority: High	**Source: Analysts**
Necessity: High	
Clarity: High	**Verifiability: Medium**
Stability:	The whole project
Description:	This means that the system shall be able to receive audio input (i.e capture customer speech via microphones) and stream the captured speech to the cloud according to the specifications provided in the Speech Recognition API.

Identifier: G-C002	
Name:	The system shall be capable of audio output.
Priority: High	**Source: Analysts**
Necessity: High	
Clarity: High	**Verifiability: Medium**
Stability:	The whole project

Description:	This means that the system shall be able to reply to the customer such as through a speaker, headphones or Bluetooth.

Identifier: G-C003	
Name:	The system shall use only approved wake words, such as “Charlie”.
Priority: High	**Source: Analysts**
Necessity: High	
Clarity: Medium	**Verifiability: High**
Stability:	The whole project
Description:	This is done so that the system is able to determine whether or not the voice stream that is receiving is directed at it.

Identifier: G-C004	
Name:	The system shall be able to provide messages based on the employee's mental state.
Priority: High	**Source: Analysts**
Necessity: High	
Clarity: Medium	**Verifiability: High**
Stability:	The whole project
Description:	This means that the system shall analyze the replies of the user and provide messages accordingly. If the user seems to be depressed the system sends messages to lighten up their mood and things to do in order to get better.

Identifier: G-C005	
Name:	The system shall be able to provide messages based on the employee's schedule.
Priority: Medium	**Source: Analysts**
Necessity: High	
Clarity: Medium	**Verifiability: High**
Stability:	The whole project
Description:	This means that the system shall send the messages when scheduled by the employee and not at random times (i.e not when the employee is busy such as in a work meeting).

Identifier: G-C006	
Name:	The system shall be able to give daily reminders.
Priority: Medium	**Source: Customer**
Necessity: High	
Clarity: Medium	**Verifiability: Medium**
Stability:	The whole project

Description:	The system must send reminders about certain tasks (i.e take a short walk, talk to a friend) that will prevent them from getting a burnout.

Identifier: UG-C007	
Name:	The user shall be able to plan an activity.
Priority: Medium	**Source: Customer**
Necessity: High	
Clarity: Medium	**Verifiability: Medium**
Stability:	The whole project
Description:	The system must allow the user to plan and schedule an activity according to their preferences.

NON-FUNCTIONAL REQUIREMENTS

Identifier: G-A101	
Name:	The system shall be secure and not expose personal information to others
Priority: High	**Source: Analysts**
Necessity: High	
Clarity: Low	**Verifiability: Medium**
Stability:	The whole project
Description:	The system shall be secure and not expose personal information to others without the user's permission. The information that is being stored is personal data and should therefore not be visible to anyone except the user itself.

Identifier: G-A102	
Name:	The system shall comply with the GDPR
Priority: High	**Source: Analysts**
Necessity: High	
Clarity: Low	**Verifiability: Medium**
Stability:	The whole project
Description:	As the system will collect data and store personal data related to the people of the European Union (EU), the data should be processed in compliance with the General Data Protection Regulation (GDPR).

Identifier: G-A103	
Name:	Charlie shall be experienced as user-friendly and self-explanatory for new users
Priority: Medium	**Source: Customer**
Necessity: High	
Clarity: High	**Verifiability: Medium**
Stability:	The whole project
Description:	Charlie should be experienced as user-friendly and self-explanatory for new users. It should be easy and intuitive to understand and

	interact with as it is important for the user's experience of the product.

Identifier: G-A104	
Name:	The system shall respond to a user within 3 seconds
Priority: High	**Source: Analysts**
Necessity: Medium	
Clarity: Low	**Verifiability: Medium**
Stability:	The whole project
Description:	When a user has a request or simply has a conversation with the VPA, the system - Charlie - must be able to respond after a maximum of 3 seconds. A response that's any later than this will have the user thinking if the system is working and there will be an unnatural flow in the conversation.

8.1.2 Study of alternative solutions

There are multiple alternative solutions that can be used in the development of the product. The following list contains the ones we have considered the most needed for the product, therefore one of these should be chosen.

- Automated Speech Recognition (ASR)
- Board computer or microcontroller
- Text to speech technology (TTS)

Automated Speech Recognition (ASR) is technology that allows human beings to use their voices to speak with a computer interface in a way that, in its most sophisticated variations, resembles normal human conversation. The use of this technology in the product will help create a more natural flowing conversation between the human user and the computer. We have compared the three most popular ASRs to find the one best suited for this project.

Board computers or microcontrollers are small computing devices with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer which we need to integrate software and sensors. We have made a comparison between Raspberry Pi and Arduino, as they are the most used ones, to find out what is most suitable for our project.

Text-to-speech (TTS) functionality is necessary as it is desired that the virtual assistant should talk, meaning that it should read the processed text in a comprehensible way. There exist several tools for this purpose. Two alternatives are Google's text-to-speech module and the Python library pyttsx3. The different alternatives will be further elaborated in section 8.1.3.

8.1.3 Valuation of alternatives

A set of parameters have been established in order to evaluate and rank the three different alternative technologies for ASR: ease of use, quality of support and ease of setup.

Table 15: Comparison of ASR technologies

ASR	Ease of use	Quality of support	Ease of setup
Deepgram	8.8	9.1	9.3
Express scribe	9.1	8.4	9.9
Kaldi	7.1	7.4	x

https://www.g2.com/categories/voice-recognition?utf8=%E2%9C%93&order=popular

To find the most suitable machine we have compared Raspberry Pi and Arduino on different features.

Table 16: Comparison of board controller vs microcontroller

	Machine learning/ computer vision	Embedded/ Low Power Projects	GPIO & Basic Electronic	Add ons & third party support	Low cost electronics/ simple circuits
Raspberry Pi	x			x	
Arduino		x	x		x

https://www.tomshardware.com/features/raspberry-pi-vs-arduino

To find the most suitable TTS we have compared Google's text-to-speech and pyttsx3 on different features.

Table 17: Comparison of TTS technologies

TTS	Quality of speech	Offline/online usage	Costs	Implementation
Google's text-to-speech	The best quality compared to other TSS	Requires a stable internet connection, cannot be used offline	Free until limit (limit: 4 million characters per month. Billing starts at 4 USD per 1 million characters)	
pyttsx3	Sound in PYTTSX is not clear and does	Can be used both offline	Free	Compatible with Python2 and 3.

	wail a little	and online		

https://circuitdigest.com/microcontroller-projects/best-text-to-speech-tts-converter-for-raspberry-pi-espeak-festival-google-tts-pico-and-pyttsx3

8.1.4 Solution selection

Based on the research and evaluation of alternative solutions we have decided to go forward with Deepgram as ASR, Raspberry Pi and Google's text-to-speech technology. We chose Express Scribe as it scores best on ease of use and also came out good on the other factors. Raspberry Pi supports add-ons and third party support which is needed for our product. Finally, Google-text-to-speech has the best quality compared to other TTS.

8.2 Use case model and traceability matrix

The codes used for the requirements in the traceability matrix are defined in section 8.1.1, and the codes for the use cases are defined in the following section, 8.3.

Table 18: Traceability matrix

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12
G-C001	x		x	x	x	x	x					
G-C002	x	x	x	x	x	x	x	x				
G-C003	x		x									
G-C004	x	x					x	x	x	x	x	x
G-C005		x	x	x			x		x			
G-C006								x	x			
UG-C007	x		x						x	x	x	x

8.3 Use cases high level description

The use cases will be assigned with Ux where x is a number.

U1: Talk to the VPA

Actor: Employee
Type: Primary
Description: The employee can have a conversation by communicating to the VPA through speech. When the employee starts their day the VPA will say "Good morning", and the employee can respond with "Good morning" as well. The VPA will also ask the employee questions during the day which the employee can

answer. It is important that the VPA understands what the employee is saying and is able to have a simple conversation.

U2: Receive personalized advice on how to take care of mental health

Actor: Employee
Type: Primary
Description: The employee will receive personalized advice from the VPA on how to take care of their mental health in general. This will include personalized advice and suggestions which in turn can help prevent burnout in the long run. The reason behind this is to raise more awareness about the importance of taking care of mental health and how to do it. The VPA can for instance suggest useful activities that the employee can do in their breaks and spare time.

U3: Plan activity

Actor: Employee
Type: Primary
Description: An employee will be able to plan an activity by communicating with the VPA. To plan an activity the user can initiate it herself and then the VPA and system will schedule it for the employee. If the VPA notices that the employee has not planned anything in the future, or it is a long time since they planned anything, the VPA can suggest to the employee if they want to plan an activity. The VPA will help schedule the activity accordingly.

U4: Adjust settings for when to get notifications from the system

Actor: Employee
Type: Secondary
Description: An employee might have preferences on when to receive notifications and this should be taken into account by the system. If the default setting for the VPA is that it asks “Have you taken a coffee break?” at 10:00, but the employee is used to having a coffee break at 09:30, then the employee should be able to change this. The employee can adjust the settings and override the default settings from the system if desirable.

U5: Customize the VPA’s voice and name to own preferences

Actor: Employee
Type: Secondary
Description: When first using the VPA the employee can choose which voice it would like (female or male), and what it wants the VPA to be called (e.g. Charlie). The default settings are gender neutral: both gender neutral voice and the name Charlie. This can be overridden whenever the employee wants.

U6: Add personalized content to the VPA

Actor: Employee
Type: Secondary
Description: The employee can add information about themselves to get even more personalized content from the system. This is information that can help the system in giving more specific content, and further advice and suggestions, towards the employee. Typical content that can be added by the employee can be their favorite song or what kind of notifications they like the most. For example, by adding their favorite song, the VPA can be set to play this song every Friday at 14:00 if the employee wants to.

U7: Answer weekly mental health check-in

Actor: Employee
Type: Secondary
Description: It is encouraged that the employee answers a set of questions asked by the VPA every week. This is a mental health check-in to see how the employees are feeling. This information can be used by the system to check if there has been any change in the mental state of the employee. This weekly check in will occur during a specific time period each week.

U8: Notify employee

Actor: System clock
Type: Primary
Description: The system clock is responsible for notifying the employee with information at specific time periods. The time periods are either default, specified by the users themselves (U4) or established based on the users work calendar (U10).

U9: Schedule appropriate time to message employee according to work schedule

Actor: System calendar
Type: Primary
Description: The system calendar will find appropriate times to message the employee. The system clock will notify the employee at the designated time slots. Messages could for instance be that it is time for a coffee break, a little office stretch etc. This is important because as a user you do not want to be disturbed in the middle of a meeting etc., so the system calendar needs to schedule an appropriate time to notify the user. Also, after a while the system will have enough information to predict when the user wants and can be notified.

U10: Plan a walk

Actor: System calendar
Type: Primary
Description: The system calendar checks for at what time it is most beneficial for the employee to take a break from work and go for a walk. What time is beneficial depends on different factors such as their meeting calendar and the employees preferred time slots to take breaks.

U11: Plan workout

Actor: System calendar
Type: Primary
Description: The system calendar checks for at what time it is most beneficial for the employee to take a break from the work and workout. What time is beneficial depends on different factors such as their meeting calendar and the employees preferred time slots to take breaks.

U12: Plan mindfulness sessions

Actor: System calendar
Type: Primary
Description: The system calendar checks for at what time it is most beneficial for the employee to take a break from the work and perform a mindfulness session. What time is beneficial depends on different factors such as their meeting calendar and the employees preferred time slots to take breaks.

8.4 Use cases prioritization

The prioritization if the use cases will be calculated be three characteristics:

a. Significant impact on architectural design
b. Includes complex, time-critical or high-risk functions
c. It represents a process of great importance in the line of business.

Values from 1 to 10 will be assigned to each use case and each criterion will e weighted to the following:

- a = 0,25
- b = 0,25
- c = 0,5

Table 19: Use case prioritization

Case of use	**a 0,25**	**b 0,25**	**c 0,5**	**Total**	**Order**
Talk to the VPA	9	9	10	9,5	1

Receive personalized advice on how to take care of mental health	5	6	7	6,25	5
Plan activity (employee)	8	5	8	7.25	3
Adjust settings for when to get notifications from the system	3	3	4	3,5	8
Customize the VPA's voice and name to own preferences	2	2	2	2	10
Add personalized content to the VPA	5	4	5	4,5	7
Answer weekly mental health check-in	2	2	2	2	9
Notify employee	7	5	6	6	6
Schedule appropriate time to message employees according to work schedule	7	7	7	7	4
Plan walk, workout, mindfulness sessions (system)	8	8	9	8,5	2

With these values and the prioritized order, we have used these considerations to decide which use cases to be solved for each cycle:

1. The closeness of the use cases in the prioritized order.
2. Use cases that build upon the finished development of other use cases will be placed in the same or later cycles. "Notify employee" is placed over "Schedule appropriate time to message employees according to work schedule" and "Receive personalized advice on how to take care of mental health" as these use cases need an implemented notification system to be finished.

According to the above, we have decided to chose 3 cycles, which will contain the following use cases:

First cycle:
Talk to the VPA.
Plan walk, workout, mindfulness sessions (system).
Plan activity (employee).

Second cycle:
Notify employee.
Schedule appropriate time to message employees according to work schedule.
Receive personalized advice on how to take care of mental health.

Third cycle:
Add personalized content to the VPA.
Adjust settings for when to get notifications from the system.
Answer weekly mental health check-in.

Customize the VPA's voice and name to own preferences.

9. Construction

9.1 First Iteration

9.1.1 First iteration analysis

Expanded format use cases description

EU1:

Use case: U1 Talk to the VPA
Actor: Employee
Purpose: To communicate with the VPA through speech.
Overview: The employee can have a conversation by communicating to the VPA through speech. When the employee starts their day the VPA will say "Good morning", and the employee can respond with "Good morning" as well. The VPA will also ask the employee questions during the day which the employee can answer. It is important that the VPA understands what the employee is saying and is able to
have a simple conversation.
Type: Primary and essential
References: *Functions:* G-C001, G-C002, G-C003, G-A104
Typical Course of Events:

Actor	**System**
1. User logs into their computer to start their work day	2. The system (VPA) will get notified and will say "Good morning"
3. User responds with "Good morning" and can continue to talk to VPA	4. Keep conversation with employee during the day, ask questions about their day

EU2:

Use case: U10 Plan a walk
Actor: System calendar
Purpose: To schedule an appropriate time for a taking a walk
Overview: The system calendar checks for at what time it is most beneficial for the employee to take a break from work and go for a walk. What time is beneficial depends on different factors such as their meeting calendar and the employees preferred time slots to take breaks.

Type: Primary and essential
References: *Functions:* G-C005
Typical Course of Events:

Actor	System
1. This use case starts when the system calendar notices the employee doesn't have any walk planned in the nearest future.	2. Check available and preferred time slots.
3. Suggest time slot for employees.	4. Schedule the walk.

Alternative Courses:

- **Line 2:** No available time slots. Not possible to schedule a walk without canceling another appointment. The operation will be canceled.
- **Line 4:** The user does not want to go for a walk. The operation will be canceled.

EU3:

Use case: U11 Plan workout
Actor: System calendar
Purpose: To schedule an appropriate time for doing a workout
Overview: The system calendar checks for at what time it is most beneficial for the employee to take a break from the work and workout. What time is beneficial depends on different factors such as their meeting calendar and the employees preferred time slots to take breaks.
Type: Primary and essential
References: *Functions:* G-C005
Typical Course of Events:

Actor	System
1. This use case starts when the system calendar notices the employee doesn't have any workout planned in the nearest future.	2. Check available and preferred time slots.
3. Suggest time slot for employees.	4. Schedule the workout.

Alternative Courses:

- **Line 2:** No available time slots. Not possible to schedule a workout without canceling another appointment. The operation will be canceled.
- **Line 4:** The user does not want to do a workout. The operation will be canceled.

EU4:

Use case: U12 Plan mindfulness sessions
Actor: System calendar
Purpose: To schedule an appropriate time for a doing a mindfulness session
Overview: The system calendar checks for at what time it is most beneficial for the employee to take a break from the work and perform a mindfulness session. What time is beneficial depends on different factors such as their meeting calendar and the employees preferred time slots to take breaks.
Type: Primary and essential
References: *Functions:* G-C005
Typical Course of Events:

Actor	**System**
1. This use case starts when the system calendar notices the employee doesn't have any mindfulness session planned in the nearest future.	2. Check available and preferred time slots.
3. Suggest time slot for employees.	4. Schedule the mindfulness session.

Alternative Courses:

- **Line 2:** No available time slots. Not possible to schedule a mindfulness session without canceling another appointment. The operation will be canceled.
- **Line 4:** The user does not want to do a mindfulness session. The operation will be canceled.

EU5:

Use case: U3 Plan activity
Actor: Employee
Purpose: To tell and plan an activity together with the VPA.
Overview: An employee will be able to plan an activity by communicating with the VPA. To plan an activity the user can initiate it herself and then the

VPA and system will schedule it for the employee. If the VPA notices that the employee has not planned anything in the future, or it is a long time since they planned anything, the VPA can suggest to the employee if they want to plan an activity. The VPA will help schedule the activity accordingly.
Type: Primary and essential
References: *Functions:* G-C001, G-C005, UG-C007. *Use cases:* U1
Typical Course of Events:

Actor	**System**
1. This use case starts with the user tells the system what activity they want to do	2. System find and suggests available time slots in the calendar
3. User choses preferred time slot.	4. System adds activity to user calendar

Alternative Courses:

- **Line 2:** No available time slots in the calendar. The user will get the option to reschedule some appointments or the operation is cancelled.

Operation contracts

Operation contract for EU1:
OC1:
Name: startSessionWithVPA(Boolean login, DateTime time)
Responsibilities: Start a session with the system
Cross-referencing:
- System Functions: G-A104, G-C002
- Use Cases:

Notes:
Exceptions: If the user starts their day without logging into their computer the VPA will not be invoked
Pre-conditions: No active session
Post-conditions: A new session with the VPA has been started

OC2:
Name: conversateWithVPA(SpeachData speechdata)
Responsibilities:
Cross-referencing:
- System Functions: G-C001, G-C002, G-C003, G-A104
- Use Cases:

Notes:
Exceptions:
Pre-conditions: A new session has been created

Post-conditions: A conversation between the VPA and the user has started

Operation contracts for EU2, EU3 and EU4:

OC3:

Name: checkCalendar(Calendar calendar)
Responsibilities: Check if the calendar is empty for activities
Cross-referencing:

- System Functions:
- Use Cases:

Notes:
Exceptions: If the calendar is full, the operation will be cancelled.
Pre-conditions: Have a calendar with missing activity.
Post-conditions: Found an available time slot.

OC4:

Name: suggestTimeSlot(Calendar calendar)
Responsibilities: Suggest time slot for the planned walk to the user

Cross-referencing:

- System Functions: G-C005
- Use Cases: U1

Notes:
Exceptions: The user doesn't want any of the suggested times. The operation is cancelled.
Pre-conditions: Available time slots.
Post-conditions: A new activity in the calendar.

Operation contract for EU5:

OC5:

Name: planActivity(Activity activity)
Responsibilities: Tell the system what activity to plan
Cross-referencing:

- System Functions: G-C001, G-C003, UG-C007,
- Use Cases: U1

Notes:
Exceptions: If there are no available time slots the system will cancel the operation
Pre-conditions: have a calendar.
Post-conditions: Possible time slots suggested to the user.

OC6:

Name: suggestTimeSlot(DateTime time)
Responsibilities: Chose a preferred time slot to do the activity
Cross-referencing:

- System Functions: G-C002, G-C005

- Use Cases: U1

Notes:
Exceptions: User do not want to chose any of the timeslots
Pre-conditions: Have available time slots
Post-conditions: New activity added to the calendar

9.1.2 First iteration Design

Sequence diagrams

EU1:

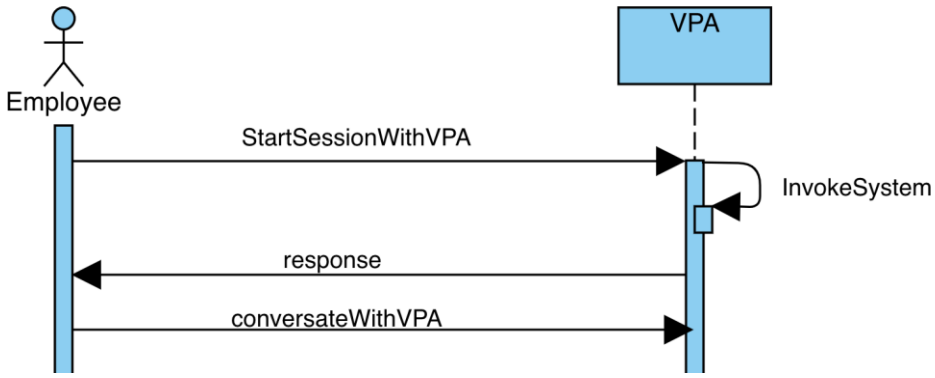


Figure 4: Sequence diagram - expanded use case 1

EU2, EU3 & EU4:

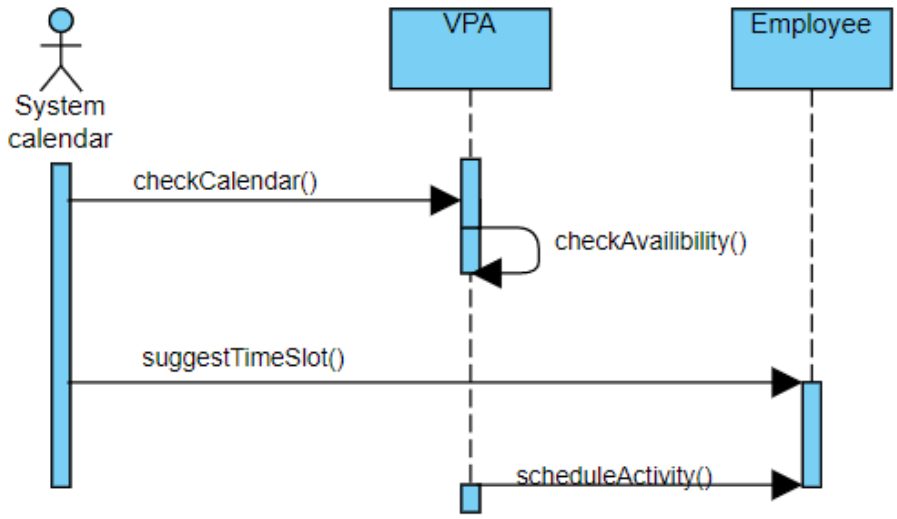


Figure 5: Sequence diagram - expanded use case 2, 3 and 4

EU5:

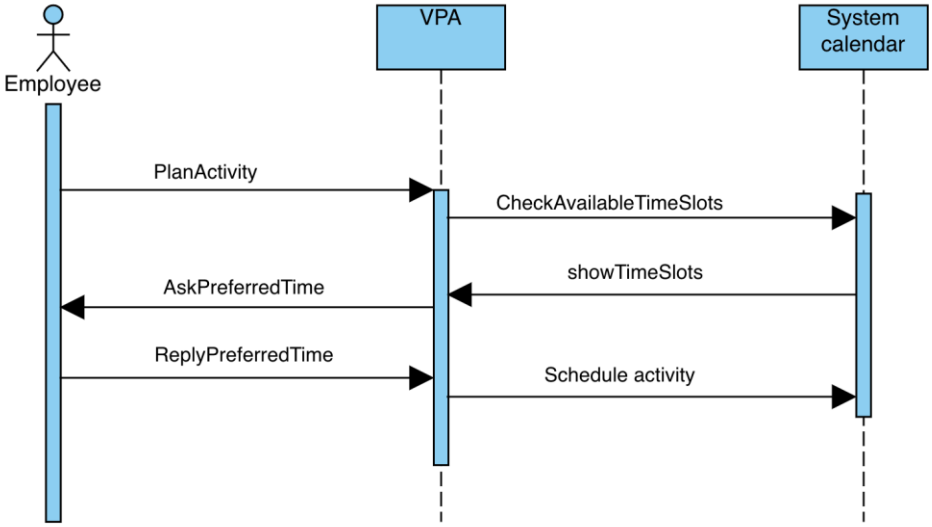


Figure 6: Sequence diagram - expanded use case 5

Class Diagram

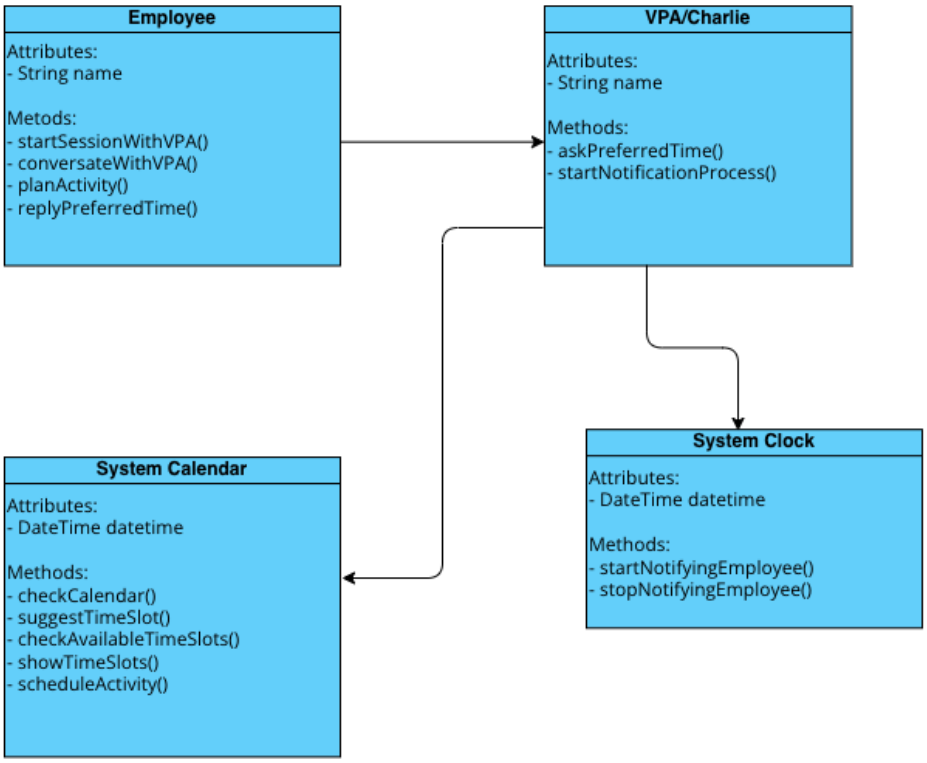


Figure 7: Class diagram

Transition State Diagram

EU1:

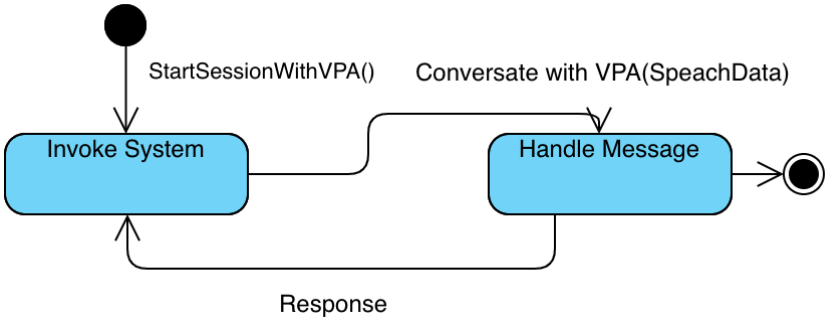


Figure 8: Transition state diagram - expanded use case 1

EU2, EU3 & EU4:

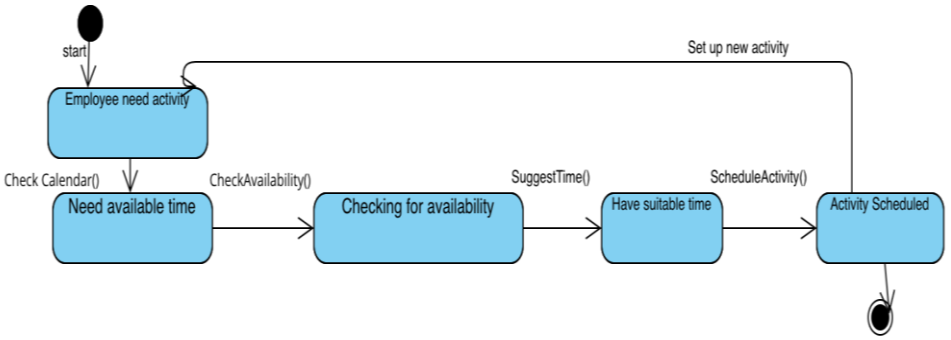


Figure 9: Transition state diagram - expanded use case 2, 3 and 4

EU2, EU3 & EU4:

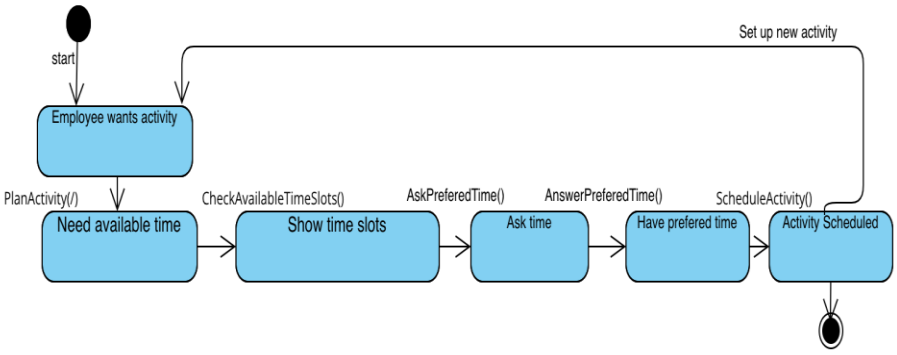


Figure 10: Transition state diagram - expanded use case 2, 3 and 4

9.2 Second Iteration

9.2.1 Second iteration analysis

Expanded format use cases description

EU6:

Use case: U8 Notify employee
Actor: System clock
Purpose: To notify the employee with information about what to do next.
Overview: The system clock is responsible for notifying the employee with information at specific time periods. The time periods are either default, specified by the users themselves (U4) or established based on the users work calendar (U10).
Type: Primary and actual
References: *Functions:* G-C002, G-C005, G-C006*. Use cases:* U4, U10.
Typical Course of Events:

Actor	**System**
2. Actor (the system clock) makes a sound every minute.	1. A scheduled notification with information is now available to the employee.
4. Actor stops making sound.	3. The system gets an input from the notification that has been received.

EU7:

Use case: U9 Schedule appropriate time to message employee according to work schedule
Actor: System calendar
Purpose: To not disturb the employee with notifications during meetings or other scheduled work events.
Overview: The system calendar will find appropriate times to message the employee. The system clock will notify the employee at the designated time slots. Messages could for instance be that it is time for a coffee break, a little office stretch etc. This is important because as a user you do not want to be disturbed in the middle of a meeting etc., so the system calendar needs to schedule an appropriate time to notify the user. Also, after a while the system will have enough information to predict when the user wants and can be notified.
Type: Primary and essential

References: *Functions:* G-C005*. Use case:* U8.
Typical Course of Events:

Actor	System
1. System calendar schedules notifications based on meetings or busy times	2. Notifies employee according to calendar

Alternative Courses:

- **Line 1**: The calendar is mistaken regarding the users available times for notifications. System will notify, and the user can postpone the message with “Charlie, send notification again in X minutes”.

EU8:

Use case: U2 Receive personalized advice on how to take care of mental health
Actor: Employee
Purpose: To give the employee advice that works with their mental health and situation.
Description: The employee will receive personalized advice from the VPA on how to take care of their mental health in general. This will include personalized advice and suggestions which in turn can help prevent burnout in the long run. The reason behind this is to raise more awareness about the importance of taking care of mental health and how to do it. The VPA can for instance suggest useful activities that the employee can do in their breaks and spare time.
Type: Primary and essential
References: *Functions:* G-C002, G-C004
Typical Course of Events:

Actor	System
	1. Processes data on the user and produces advice and activities based on the perceived mental state of the user.
3. When the employee is ready to hear the notification, they will say “Charlie, read message” or other personalized receive message.	2. Send notification with advice to the user.
	4. Reads message

Alternative Courses:

- **Line 4**: The user doesn't hear the message or wants to hear it again and says “Charlie, replay message” to hear it once more.

Operation contracts

Operation contracts for EU6:

OC1:

Name: startNotifyingEmployee()
Responsibilities: Provide a notification to the user, regardless of what the notification is about
Cross-referencing:
- System Functions: G-C002.
- Use Cases: U2, U8

Notes:
Pre-conditions: A message is available for the user to receive.
Post-conditions: Sound is made.

OC2:

Name: stopNotifyingEmployee()
Responsibilities: Stop notifying the user that a message is available.
Cross-referencing:
- System Functions: G-C002.
- Use Cases: U2, U8

Notes:
Exceptions:
Pre-conditions: A notification sound is being made every minute.
Post-conditions: System alarm stops making notification sound.

Operation contracts for EU7:

OC3:

Name: scheduleNotifications (DateTime time)
Responsibilities: Schedule a notification for the users only at the time when they are not busy (according to their calendar schedule).
Cross-referencing:
- System Functions: G-C005
- Use Cases: U8

Notes:
Exceptions: The calendar is mistaken regarding the users available times for notifications. System will notify, and the user can postpone the message with “Charlie, send notification again in X minutes”.
Pre-conditions: The system wants to create a new notification for the user.
Post-conditions: New notification is scheduled in the user’s calendar.

OC4:

Name: readMessage(Message message)
Responsibilities: Read message when receiving confirmation from the user.
Cross-referencing:

- System Functions: G-C002, G-C003.
- Use Cases: U2.

Notes:
Exceptions: If the user doesn't hear the message or wants to hear it again and says “Charlie, replay message” to hear it once more.
Pre-conditions: Notification is associated with a message.
Post-conditions: Message has been read.

Operation contract for EU8:
OC5:
Name: confirmNotification()
Responsibilities: Confirm to the VPA that the employee is ready to hear the message
Cross-referencing:
- System Functions: G-C001, G-C003
- Use Cases: U2.

Notes:
Exceptions: If a message is already received, the user can say “Charlie, please replay message” to hear it once again
Pre-conditions: Notification is associated with a message.
Post-conditions: Message has been read.

9.2.2 Second iteration Design

Sequence diagrams

EU6:

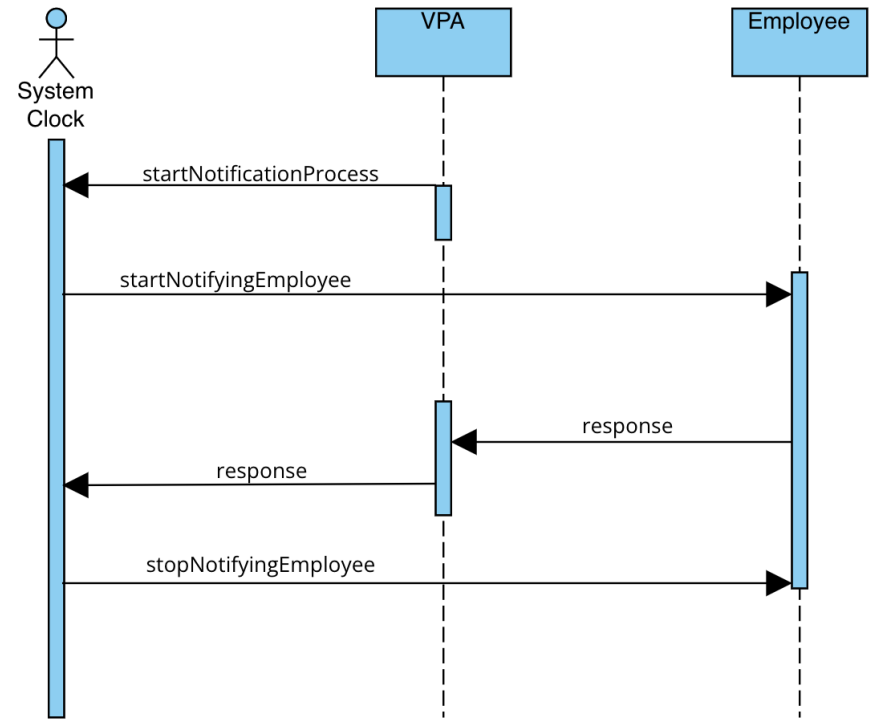


Figure 11: Sequence diagram - expanded use case 6

EU7:

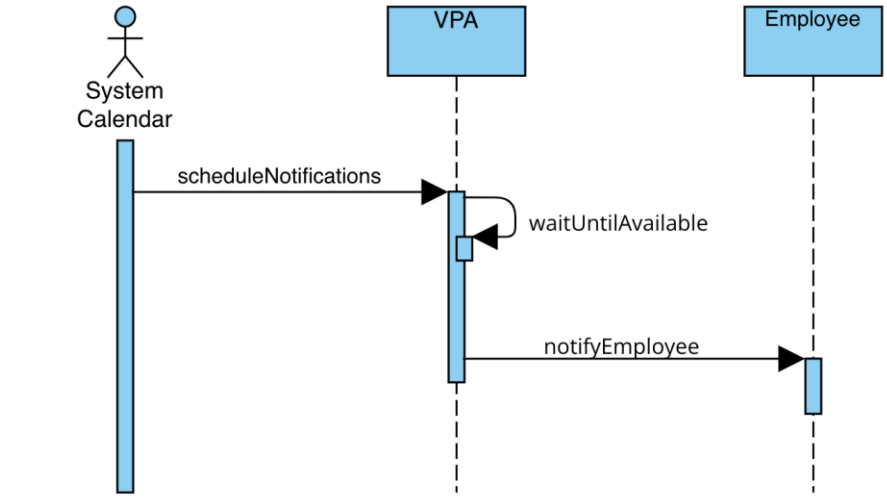


Figure 12: Sequence diagram - expanded use case 7

EU8:

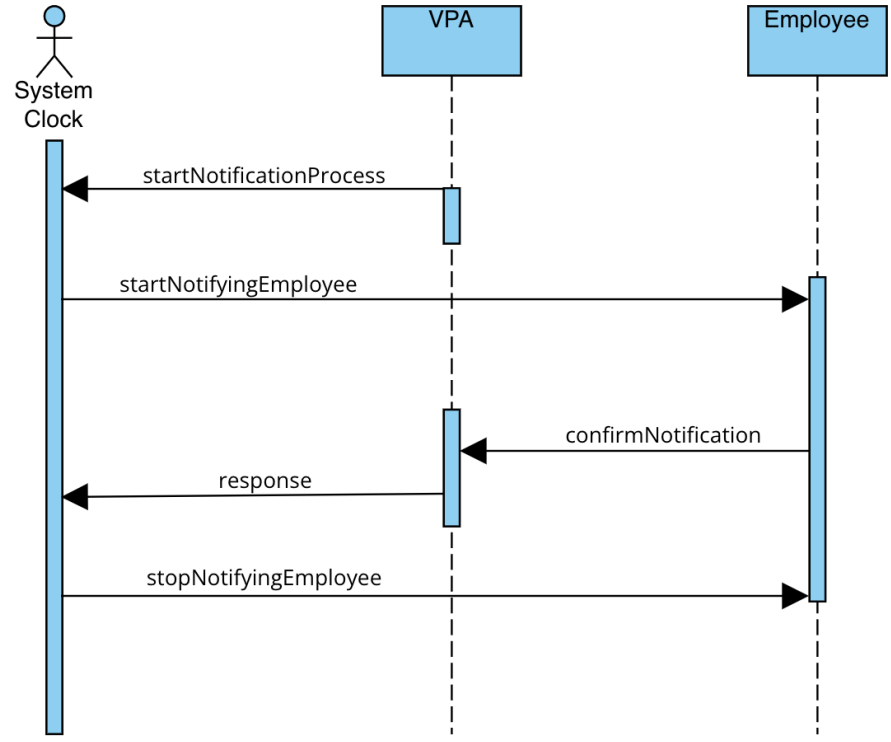


Figure 13: Sequence diagram - expanded use case 8

Transition State Diagram

EU6:

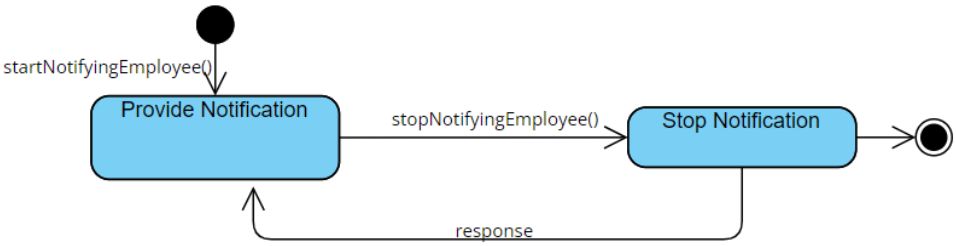


Figure 14: Transition state diagram - expanded use case 6

EU7:

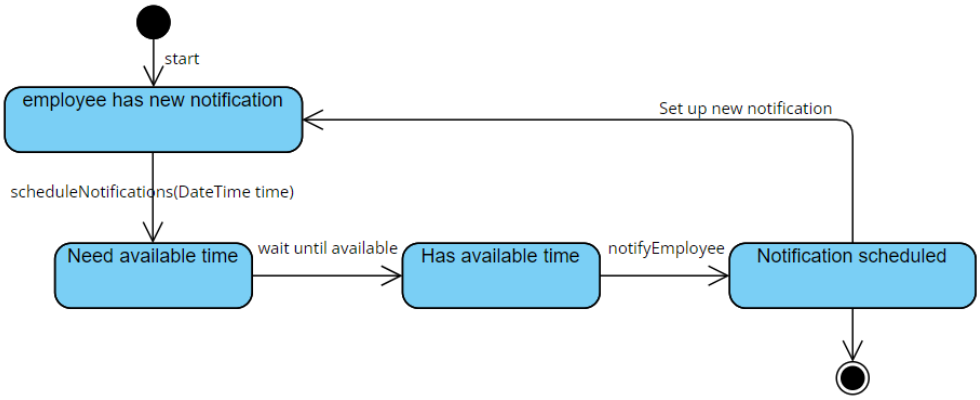


Figure 15: Transition state diagram - expanded use case 7

EU8:

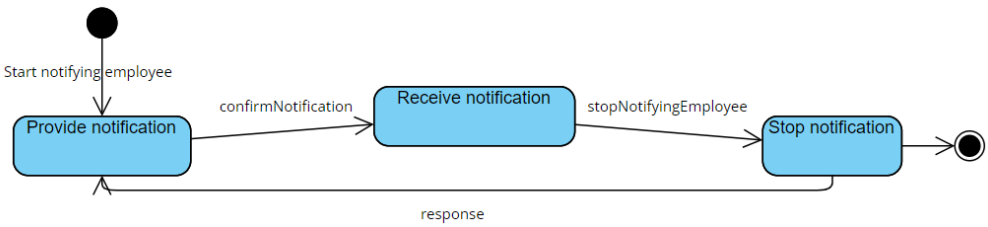


Figure 16: Transition state diagram - expanded use case 8

10. Execution of the quality plan

This information can be found in section 4.10.

11. Execution of the configuration management plan

This information can be found in section 4.11.