EPOKA UNIVERSITY

BlueCollar

CEN302- Software Engineering

02/06/2023

CEN302- Software Engineering

Contents

Problem description and solution

By doing a thorough research and analysis of challenges of our daily life we came up to the conclusion that one problem that has always been an obstacle was finding a handyperson in the moment without having to ask our friends, neighbors, or relatives for a "known guy" that would offer us help. Handymen provide us with an easy-to-use service that is completed quickly and efficiently. A handyman is well-versed in a wide range of tasks and repairs. Their primary responsibilities include repairing plumbing systems, repairing company equipment or tools, and testing various company or home appliances to ensure proper operation.

The purpose of this project is to develop an application that will allow us to solve a problem we encounter in real life or provide a service through which we can facilitate the service sector. Keeping this in mind we thought that a Handyman Service Web App would be the ideal solution to offer these services with only a click away.

Our web application would offer a variety of services such as renovation, electrical services, plumbing and so on. Also, the consumer would have the option to set up a personal account to keep track of the services they require and utilize. The customer might schedule an appointment and, if necessary, make a payment.

The online application would provide a section exclusively for handy people so they could manage their appointments, add or remove features from their profiles, and make other necessary changes.

All told, we shall establish two sides, one that provides a service and one that accepts it as required.

Aim and main objectives

The aim of this project is to create a website which fulfills both customer and service providers (handymen) requirements. The website should be:

- well designed
- functional
- user-friendly
- have a responsive design.

Objectives

- Engagement: giving users a reason to return, use the app's features, and take the desired action.
- Set a clear idea of who our target audience is, if we want to make the most of our business website potential.
- Feasible: the web app should be built in a way that fits well with particular needs, activities, and plans.
- Making a good first impression and clearly communicate to the customers why our website exists in the first place.
- Pick one or two actions we want potential customers to take on our website: make it simple to contact our company (call, arrange an appointment, etc.) and include contact information on every page of the website.
- Maintaining a straightforward and uncomplicated navigation bar. Having too many navigation options will make website visitors confused.
- Make it simple to get back "home." Visitors desire a simple way to get back to the homepage.

Description of the application

This application is designed to bring together handypersons and customers in need of repair work. This includes problems with a wide variety of electric appliances and house repair problems (water or electricity for instance). The application is intended for use in the Albanian region.

The application is web based, operates on mostly any web browser and operating system and requires internet connection. The application first displays a greeting and prompts a log-in or sign-up form. When signing up you need to specify your profile as either a customer or handyman. The customer signs up with their first name, surname, email, contact information, payment details and address. The handyman profile will require: name, surname, city, payment information, contact information, email, service area, working hours and pay rate. Once the customer has signed up first and logged in, he/she is able to use the functionalities of the application. The user will be displayed a list of handymen with their respective information. The user is able to schedule the services of the desired handyman and make the payment online if desired. On the other hand, when logged in as a handyman providing services, you will be displayed your appointments including the payment status. You may as well edit your information details as needed. The access of these respective functionalities will be with buttons and information will usually be displayed in table format, entrance of information and editing will be done using forms. The application will be build using HTML, CSS, JavaScript, PHP and MySQL for database connection needs. As seen fit, the appearance design will be easy on the eye and response time will be adequate. The application aims simplicity and ease of use.

Requirements Elicitation and Analysis

Requirements Discovery

This process is known as "The process of gathering information about the required and existing systems and distilling the user and system requirements from this information."

In order to discover the requirements we need to identify the stakeholders. As we know a stakeholder is an individual or institution that has an interest or concern in the application being developed by our team.

The stakeholders can be divided into internal and external stakeholders.

Internal stakeholders:

- Software developers
- Client engineers
- System architects
- Owners/Shareholders of the company that develops the system
- Project Manager who manages and supervises the application development
- Executives who make critical organization-wide decisions

External stakeholders:

- End-users who will use the system directly
  - The handyman who will register themselves
  - The clients who will find the handyman's services
- Customer who will purchase the application
- Government/ Regulatory agencies whose legislation and rules might impose restrictions
- Suppliers who supply our company with different hardware and software
- Competitors who view our work for business competition purposes

We need to interact with stakeholders in order to acquire information which will help us distil the user and system requirements. This can be done through different methods and we will be using interviewing. The interview will consist of a mix of closed and open questions. These questions were made to stakeholders with the respective common answers which depict the requirements:

These questions were made to stakeholders with the respective common answers which depict the requirements:

1. What features should the application provide for the handyman?

Stakeholder	Handyman	Engineers
1.	I would like for appointments at home to be made online and to be able to view my appointments.	The handyman should be able to edit any information that they enter
2.	I would like to cancel an appointment made or edit it if a personal problem came up for me.	
3.	I would like for the payment to be made online from the customer in order not to have any problems and to be more transparent.	
4.	I would like to post my details in a wall post, with the essential information to contact me, leave an appointment.	

- The handyman should be able to create a profile in the application.
- Be able to edit their profile information.
- Be able to list their service details for viewing.
- Be able to edit their service details.
- Be able to view appointments made, edit, decline them if wanted to do so.
- Be able to view payments made for the service.
- Be able to view the ratings with reviews.

2. What features should the application provide for the client?

Stakeholder	Client	Engineers
1.	I would like to create an account in this application.	The client should be able to edit any information that they enter, in profile.
2.	I would like to view the different handyman detail posts or listings, in small type rectangles with the essential information and to have buttons there for me to click to leave an appointment or make a review.	
3.	I would like to be able to leave a rating for the handyman's service with a 0 to 5 scale.	
4.	I would like for there to be different categories of handymen at the head of the web application so I can directly navigate to the listing that I want.	
5.	I would like to cancel, or edit the appointments if I change my mind.	
6.	I would like to view all appointments that I have made.	
7.	I would like to make the payment online in order to avoid awkwardness or sudden price changes.	

- The client should be able to create a profile in the application.
- Be able to edit their profile information.
- Be able to view the different handyman listings.
- Be able to make an appointment with a handyman.
- Be able to view the personal appointments made.
- Be able to cancel, edit an appointment.
- Be able to make the payment to the handyman.
- Be able to rate the handyman and leave a review.
- There should be different tabs for different handyman service categories.

3. Who will know that an appointment or payment is made? (Asked to the client of the app) Only the involved users, other handymen or users will not have any information.

4. What information should the profile of the handyman include when signing up? (asked to the handyman)

- ✓ Name, surname.
- ✓ Phone number, email address.
- ✓ City
- ✓ Service area.
- ✓ Payment information.
- ✓ Working hours.
- ✓ Pay rate.

5. What information should the profile of the client include when signing up? (Consulted individuals and developers)

- ✓ Name, surname.
- ✓ Phone number, email address.
- ✓ City.
- ✓ Home address.
- ✓ Payment information.

Questions not predetermined and that came up during the interview:

6. Should other clients be able to find other client's profiles? (Asked the client)

Since this is not a social media type application there is no need for other clients to find other clients since the objective of this application is to connect handymen to clients.

Stakeholder	Client
1.	I would like not to be able to find other clients since this is not Instagram.
2.	I would like for the app to be simple.

7. Should a handyman be able to view other handymen? (Asked the handyman)

Yes, the handyman can see other handyman's listings for competition purposes where one can edit their pay rate information for instance.

Stakeholder	Handyman
1.	I would like to see who my competition is.
2.	I would like to see how much other people are paying for services similar tomine so I know when to increase or decrease the rates.

8. Can a client cancel a payment made to a handyman? (Due to budgetary, time or contract constraints with the banks, engineers have consulted it is not available at the moment) No, you cannot do this through the application and contacting banking institutions may be of help and is suggested.

9. Are there any government regulations to the price of the pay rate? (researched the government of Albania's tax law)

Yes, it should include an additional 20% of the original price in order to include the tax.

10. Where is the intended region for use? (Consulted by the owners and project manager) The territory of Albania at first, if successful enough, may be considered internationally.

11. Should the information posted by the handyman as a job posting include the ratings reviews or are they personal to the handyman? (Asked some individuals)

The ratings should be included in the handyman postings, but the reviews can be shown when you try to click the full profile of the handyman. The clients would not like for the page listings to be too cluttered with too much information.

Stakeholder	Client
1.	I do not want the post listings to be too
2.	I would rather have the reviews be shown when I click on the handyman's detailed information if I would be interested in it.

This way we can depict:

The application domain is a web-based application connecting home-based B2C services. So it operates online in a laptop, smartphone or tablet.

Initial requirements and services the system should provide in an informal form derived from the interview:

1. The application should allow the handyman to sign up and log in with the respective information.
2. The handyman should be able to post their information details.
3. The handyman should be able to view their appointments.
4. The handyman should be able to cancel, edit their appointments.
5. The handyman should be able to view the payments made to them.
6. The handyman should be able to view their rating and review.
7. The handyman should be able to edit their information posted.
8. The handyman should be able to view other handyman listings.
9. The handyman should be able to use tabs for the services to navigate for ease of use.
10. The application should allow the clients to sign up and log in with their credentials.
11. The clients should be able to view the different handyman posts.
12. The clients should be able to make an appointment with a handyman.
13. The clients should be able to view the appointments made.
14. The client should be able to make the payment to the handyman.
15. The client should be able to cancel, edit an appointment.
16. The client should be able to rate and leave a review to the handyman.
17. The client should be able to edit their information.
18. The client should be able to use tabs for the services to navigate for ease of use.

Requirements classification and organization

We can group the related requirements and organize them into coherent clusters according to how relevant they are with each other.

According to the functionalities:

Sign-up and log in requirements:

- ✓ The application should allow the handyman to sign up and log in with the respective information.
- ✓ The application should allow the clients to sign up and log in with their respective information.

Information details requirements:

- ✓ The handyman should be able to edit their information posted.
- ✓ The client should be able to edit their information.

Handyman job posting requirements:

The handyman should be able to post their information details.

- ✓ The clients should be able to view the different handyman posts.
- ✓ The handyman should be able to view other handyman listings.
- ✓ The handyman should be able to use tabs for the services to navigate for ease of use.
- ✓ The client should be able to use tabs for the services to navigate for ease of use.

Appointments:

- ✓ The handyman should be able to view their appointments.
- ✓ The handyman should be able to cancel, edit their appointments.
- ✓ The clients should be able to make an appointment with a handyman.
- ✓ The clients should be able to view the appointments made.
- ✓ The client should be able to cancel, edit an appointment.

Payment:

- ✓ The handyman should be able to view the payments made to them.
- ✓ The client should be able to make the payment to the handyman.

Rating and review:

- ✓ The handyman should be able to view their rating and review.
- ✓ The client should be able to rate and leave a review to the handyman.

We also group and organize according to the clusters of profile user:

Handyman:

1. The application should allow the handyman to sign up and log in with the respective information.
2. The handyman should be able to post their information details.
3. The handyman should be able to view their appointments.
4. The handyman should be able to cancel, edit their appointments.
5. The handyman should be able to view the payments made to them.
6. The handyman should be able to view their rating and review.
7. The handyman should be able to edit their information posted.
8. The handyman should be able to view other handyman listings.
9. The handyman should be able to use tabs for the services to navigate for ease of use.

Clients:

1. The application should allow the clients to sign up and log in with their respective information.
2. The clients should be able to view the different handyman posts.
3. The clients should be able to make an appointment with a handyman.
4. The clients should be able to view the appointments made.
5. The client should be able to make the payment to the handyman.
6. The client should be able to cancel, edit an appointment.
7. The client should be able to rate and leave a review to the handyman.
8. The client should be able to edit their information.
9. The client should be able to use tabs for the services to navigate for ease of use.

Prioritization and negotiation

At a total view of the initial requirements we do not view conflicts between the requirements. We can prioritize the initial requirements according to the clusters that we have depicted above, as such:

1. Sign-up and log in requirements because they are fundamental, without signing up and logging in we cannot access the application.
2. Handyman job posting requirements because it is important to the basis of the application to have the job postings or information details posted.
3. Appointment requirements are next because it is important to allow a client to make an appointment with a handyman, cancel, edit and for the handyman to view them and interact with them the same.
4. Information detail requirements are next because the handyman should be able to change their information details if needed when there is a change or when they see that their rates are too high or low and a price change is needed meanwhile for the client when the real life information changes and wants to keep the details updated.
5. Rating and reviews are important for the legitimacy of the handyman and their services provided, if other clients had a positive or negative experience and help with the decision making of the client of whether or not to choose that handyman.
6. Payment requirements are the last priority because the handyman will after all go to the client's house and meet them and as such there is a second chance for a payment opportunity.

Requirements Specification

User and System Requirements

1. The application should provide a sign up or login form for users (clients and taskers) in order for them to get the services that the app provides after creating an account or logging in to the existing one.
    a. The first page of the application should have the login and sign up button.
    b. For every user who has already an account the data that he provides when logging in should match the one that is previously stored in the database.
    c. For the new users the sign up window should be provided where they should fill in required information like name, last name, personal number or email address.
    d. When clicking the login and sign up button the user should be redirected to a new window where a form created using php will appear.
    e. When we create a new account the user will receive an email of configuration about the new account created.

2. The application should have general information listed in the navigation bar.
    a. The moment that the user opens the application (after logging in), they will be greeted by the main/home page.
    b. The home page would hold a difference between the taskers and the clients. The client's home page will include about us, services section, meanwhile the taskers home page will include about us, appointment and wallet section (text area).
    c. We are going to use Submit buttons that would redirect the user to different locations within the app using either HTML form tags (action redirecting to another php page) or href when dealing with links.
    d. The information written would be static in the about us and services section, only changeable by the software engineer, while the appointment, reviews and wallet would be dynamic changing whenever the user or tasker clicks to either of them.

3. The user must be able to make payments for the services they require, and the taskers must be able to get paid when they get booked.
    a. After logging in, creating an account the user will be able to pay for every service.
    b. The user will be asked for the transaction requirements after they have filled the required fields in the booking form.
    c. Create an HTML form to collect the customer's payment source information.
    d. Create a token to transmit the encrypted data securely.
    e. The next step is to submit the form with all relevant data to the customer's source

bank for the requisite approval.

f. Once the customer's source bank confirms the transaction, verify and validate the charges involved in the order process.
g. Whether the transaction is approved or rejected by the customer's source bank, display the transaction's status to the consumer.
h. The payment details need to be stored in the database.
i. A HTML form will be visible for the handyman of every payment they have received and their current summed up balance.

4. The handyman profile should have lots of dynamic features that would allow them to change their description whenever they like.
    a. Since the handymen would be the ones to provide the service they should be able to have more features in their profile that would help change said services.
    b. When signing up the user would have the opportunity to choose among two options: that of being the service provider or the service taker. When the user chooses the handyman option they will be given additional features automatically as they log into their account.
    c. The handyman should be able to change their profile information as many times as they want (dynamic system).
    d. These changes will be done by clicking on the area they want to change or clicking the pen symbol to the right of the text area.
    e. Every change that the tasker does will automatically be updated to the database.
    f. The page will not require a manual refreshing of the data but it will automatically change.

5. The users must be able to leave reviews and they should be visible in the taskers profile.
    a. Every handyman should have in their profile a text area where users will be able to leave their reviews and ratings as they prefer.
    b. These reviews, after being submitted by the user, will be saved to the database of the handyman. Their database will be updated every time a review is being written.
    c. The handyman profile will include a HTML table with every review that will be public for everyone accessing their profile.

6. Users must be able to book, cancel or change appointments.
    a. The user must be able to book an appointment with the handyman.
    b. To make this process easier we must display in the handyman profile whether they are free or booked at a specific time and day by accessing the database of the tasker and displaying every change.

c. When the handyman is booked they should have an updated HTML list of their booking time and day in their profile.
d. The handyman must be able to check by clicking a certain appointment request whether the time that a client wants to book is available or not. If that time is not available the tasker will click the “reject” button and the user will get an email/message that their request has been rejected and that they need to book at another time.
e. If the appointment is accepted (the tasker has clicked the “accepted” button) but the user wants to cancel it they must be able to do so, by clicking the booked appointment in their appointment list (displayed in their profile) and selecting the cancel button. The database of the handyman where this appointment was stored will be updated and the appointment will be deleted from the database automatically.
f. If the appointment time and date is accepted by the handyman but this schedule is not convenient anymore for the client then they must be able to change it by clicking said appointment. These changes will be reflected to the database of the handyman and the client, which will then lead to changes in the HTML forms displayed in their profiles.

7. Classification of the services that the app provides and all the taskers in every area of each service.
    a. The home page should include a classification button that when pressed redirects the user to another php page that includes all various services that we provide.
    b. When the user clicks to a certain service they will be redirected to another php page that includes the list of all handymen that provide this service.
    c. If moving further when clicking the name of a certain service provider the button will redirect to their profile page.
    d. The client will be able to view the taskers profile with every feature mentioned above.

Functional, non-functional and domain requirements

An effective software system is supported by both functional and non-functional criteria. Each criterion has distinct qualities that improve usability, functionality, accessibility as well as security by enabling software to operate more quickly and employ robust security procedures.

Functional requirements are requirements that the end user specifically requests as basic system facilities and describe ways a product must behave. In simple words these requirements outline what a software system should or should not do and they also provide a summary of the features and functionality that the system has.

There are several types of functional requirements that can be part of the software system such as:

1. Operations and workflows the system must perform
2. Formats and validity of data to be input and output by the application
3. User interface behavior
4. Data integrity and security
5. Safety and regulatory measures
6. Validation of user access/authorization for use and modification (create/modify/delete) of the system.

FUNCTIONAL REQUIREMENTS [1]

Requirement Statement	**Must/Want**	**Comments**
1. System must allow new members (clients & handypersons) to sign up and login by providing to them all the needed data fields.	MUST	Form + data stored in DB
2. System must allow existing users to directly login with their saved credentials.	MUST	
3. The system must send a confirmation email when a *new* user account is created.	MUST	
4. The system must allow users to reset their password by clicking on *"Forgot Password?"*.	MUST	
5. The system must enable users to add, delete, or modify their data and information by clicking on *"EDIT"*.	MUST	
6. The system should allow only managers to access *insensitive user's data.*	MUST	

7. The system must allow users to easily navigate from the home page through the other sections. • The website should have a *USER* homepage where the following headers can be found: about us, services, profile. • The website should have a *Service Provider* homepagewhere the following headers can be found: about us, appointments, wallet, profile.	MUST	
8. The system shall provide a list of workers in accordance with the relevant service categories alongside with their respective information.	MUST	Redirect user to a new page that will display information on the availability, area of service and fees.
9. System should allow users to view details and information about each worker.		
10. The system must enable users to leave comments and ratings in the appropriate section.	MUST	*Review Section* features
11. The system should to make it possible for the user to set up an appointment.	MUST	Redirect user to the *booking form* after clicking the **"Book me"** button.
12. System shall allow users to change the status of their appointments by cancelling, postponing or changing.	MUST	
13. The system must enable and allow the user to make the payment according to the available options: • Pay when the service is received or • If the user chooses on the *"Pay now"* option he will now be redirected to the chosen payment platform which will enable payments between parties through online money transfers.	MUST	The user must insert his credit card information: *card number, expiry date, cardholder name and the card's authorized 3-digit code (CVV2 security code).*

14. System should allow users' payment to be displayed on the handyman's wallet information.	MUST	
15. System should allow users to checkout using saved account data.	MUST	Approve and complete payment transaction

NON - FUNCTIONAL REQUIREMENTS

A non-functional requirement is a description of the nature of the object, its construction, or a constraint on its design or behavior. In short, these define how well it will function, including here aspects like performance, security, dependability, data accuracy and so on.

As was previously stated, these requirements must be built upon the following standards:

1. Usability: How simple is it for users to operate and use this application?
2. System Reliability, Maintainability and Availability: How often does the system experience critical failures? When a problem develops, how long does it take to fix it? And how is user availability time compared to downtime?
3. Scalability & Performance: How quickly does the system generate results? How much will this efficiency alter as workloads increase?
4. Security: How well are the data and the system secured against intrusions and cyber- attacks?
5. Localization: Is the system compatible with local specifics?

NFR Documentation

1. It should take less than 1.5 seconds for the application's home screen to load.
2. The landing page must respond in 6 seconds or less in a Chrome desktop browser, rendering text and images over an LTE connection included. This website must serve 800 users per hour.
3. All website pages should load within 3 seconds with the total number of simultaneous users being less than or equal to 1500.
4. The system must be scalable enough to handle, accommodate and support 8,000 simultaneous visitors while still maintaining an optimal performance.
5. The system runs on Windows 10 and must be able to operate on Windows 11 without experiencing any performance or behavior changes.
6. During a month, the system must operate effectively and efficiently in 95% of use scenarios.
7. Following a system breakdown, the mean time to restore the system cannot be longer than ten minutes.
8. Users must have access to the web dashboard 99.98% of the time each month during EST work hours.
9. The system will demand that the user abide by a number of guidelines or limitations established for the password field, including: length, letter, characters, and number entered.
10. The following format is required for date and hour respectively: month. date. year; hour. minutes. seconds.
11. The percentage of users who enter incorrect payment information on the checkout page cannot be higher than 10%.

Domain Requirements [3]

Domain requirements reflect the environment in which the system operates. They are important because they often reflect fundamentals of the application domain. If these requirements are not satisfied, it may be impossible to make the system work satisfactorily.

- ✓ ***Payment Gateway Regulations*** [2]
  - Since our application will include only payment we feel it is our responsibility as service providers to take all reasonable precautions to prevent the theft of personal data during user-to-user transactions.
  - The payment card industry data security standard, or PCI DSS for short, is one approach to check if the payment gateway complies with all the regulations and standards.
  - PCI DSS consists of a standardized, industry-wide set of rules and procedures for various security controls.
  - It was created to protect credit card and cardholder data (CHD) and is applicable to any business that gathers, processes, stores, or transmits this data.
  - The PCI Security Standards Council (SSC), which is the organization in charge of enforcing PCI DSS, is composed of the largest credit card firms, including MasterCard and Visa.
  - The PCI DSS's goal is to guarantee that card payments are subject to the proper safeguards, and the first step to accomplishing that goal is to finish an assessment (the specifics depend on your level), a quarterly network scan, and the Attestation of Compliance Form.

- ✓ ***Privacy Policy & Data Protection:*** The General Data Protection Regulation (GDPR) is a legal framework that sets guidelines for the collection and processing of personal information from individuals who live and outside of the European Union (EU).

- ✓ ***Handyman's license:*** The license is a necessary and mandatory document for verifying the employees' skills in the service sector they operate in.

- ✓ ***NDA (Non-Disclosure Agreement):*** is intended for protecting confidential information between parties, in this case between the organization and service providers.

- ✓ ***Terms & Conditions:*** TC is intended to protect the organization. They allow business owners to set their own rules (within the bounds of applicable law) for how their service or product may be used.

- ✓ ***App Stores Requirements:*** All requirements for app publishing established by Google and Apple guidelines must be met by mobile apps. They emphasize the importance of protecting personal data, such as health information and information obtained from minors, as well as intellectual property concerns regarding your mobile device.

Model Development

We have selected the waterfall method with a linear-sequential life cycle model to design our application. This model is really straightforward, simple to comprehend and keeps the focus on the end goal at all times. There is no overlap between phases in a waterfall model; each step must be finished before the subsequent phase can start. For the application to be designed properly, it is crucial that we have a comprehensive grasp of the user's expectations.

We chose this methodology since we are documenting our project in stages. Each phase is handled by thoroughly researching the features that must be included in the application and consulting with the client about what they need and want to see there. This methodology is more convenient for us to use because it enables us to focus solely on creating the software design with all of the features included and putting it into use after carefully going through each customer request and creating comprehensive requirement documents.

The Waterfall method can be the best choice if the project is small, the technology is well-understood and developed, and the requirements are clear, well-defined, and well-documented. Also, budget, timeline, and scope outcomes may be more predictable as a result of this approach.

Requirements Validation

The main issue and challenge in developing an application is “Are we building the right system?” and the answer to this question can be given by validation. Requirements validation is the process of confirming that the requirements we have given are appropriate for development and specifying the system that the client actually desires. Requirements validation will assist us in spotting problems early on in the application development process and avert the need for unnecessary rework later on in the system development life cycle. Additionally, we will verify that the system contains all relevant requirements components and that the published requirements adhere to the expectations of stakeholders through requirements validation.

Requirements Checking

With requirements checking, we will determine whether the system meets the needs of the user and has all the necessary features, whether there are any requirements conflicts, whether we can implement the requirements given the budget and technological constraints, and whether we can check all of the requirements.

Providing functions that support the user's needs necessitates careful evaluation of the user's wants and preferences, as well as the system's capabilities and limitations. The requirements and functions mentioned in the user's specifications, such as having general information about the services, booking, cancelling, and changing appointments, and so on, do meet all of the user's needs. We cannot fully prevent conflicts; however, we hadn’t had any requirements conflicts with either stakeholders or users regarding the development of the application up to this point. A significant consideration for our application is the feasibility of implementing requirements given the available budget and technology. We performed a feasibility study that considered the cost ($1000) and resources needed to implement the system. We estimated the time, money, and resources needed to execute each requirement and compared them to the available budget and resources. We can state that the system meets the available budget and technological constraints by taking a realistic and collaborative strategy.

Requirements checking are an important part of the requirements development process, as it ensures for us that the requirements are clear and complete before we begin with developing the application. In this stage we review and validate the requirements to ensure that they meet the needs and expectations of the stakeholders, and users, and can be implemented within the available time. There are several techniques we can use to check the requirements, but we continued with automated testing tools to help us identify any ambiguities, inconsistencies, or errors in the requirements, as well as any gaps or omissions that may need to be addressed, before it’s released into production.

Requirements Validation Techniques

Requirements reviews

During requirement review, we assembled a team of individuals from the user and organizational sides to carefully examine the software requirements specifications and evaluate them in detail in order to spot any potential issues. The reviews then are used for quality assurance and data gathering in the software development. The review group consists of the author of the required documents, someone who knows the user's needs a member of the design team, and the individual or individuals in charge of maintaining the requirement document. Throughout the review process, the reviewers will look for inconsistencies, disputes, or omissions in the requirements and make suggestions for changes or revisions to make them more comprehensive and feasible. The review method is collaborative and constructive, with an emphasis on improving the requirements' quality and effectiveness of our application.

Review Checks

We perform review checks after concluding the requirements review technique to ensure that the requirements are testable, properly understood, the origin of the requirements is clearly stated, and the requirements can be changed without having a significant impact on other requirements.

Requirements management

Requirements management is used to ensure that the objectives of product development are effectively attained. A number of techniques have been developed for documenting, analyzing, prioritizing, and agreeing on requirements in order to guarantee that the engineering teams always have approved and up-to-date requirements. Requirements management provides a way to avoid mistakes by keeping track of requirements changes and encouraging communication with stakeholders throughout the project's engineering lifecycle. The product manager is typically responsible for curating and defining these requirements. However, clients, partners, salespeople, customer service representatives, management, engineers, and members of the operations and product teams are just a few of the stakeholders who can develop specifications. Constant communication is necessary to ensure that the engineering team is aware of changing objectives in order to have the best end result.

Requirements management plan (RMP)

An RMP (requirements management plan) aids in describing how we will receive, analyze, document, and manage every requirement for a project. An essential part of a requirements management strategy is defining the project overview, requirements collection process, roles and duties, tools, and traceability.

In our case, the typical plan entails developing the application so that users can schedule handyman appointments in accordance with their needs. It also includes more specific product requirements that may be gathered throughout a project's lifecycle, such as what the application should exactly include, how users will be able to book appointments, how payments will be made, how to create a feedback area, and so on.

Requirements management process

A typical requirements management process complements the systems engineering V model through these steps:

1. Collect initial requirements from stakeholders
    - Communicating with stakeholders and creating an idea of what exactly the application is about.
    - Getting all the information needed of what should be included in it.
    - For instance the stakeholder asks to create a feedback area where the users leave their comment regarding the service they got.

2. Analyze requirements: After gathering all the needed information from the stakeholder, we review the requirements and start analyzing them and how we could adopt them in the application.
3. Define and record: We create a document where all the requirements the stakeholders asked for are listed
4. Prioritize requirements: We share the document with the stakeholders, and then they let us know if we should move on with implementing those requirements or if they want to make changes to them
5. Agree on and approve requirements: After agreeing on the requirements listed, we start to implement them in the application
6. Trace requirements to work items: When the implementation is done, we trace its functionality.

Since we are using a waterfall method there will be no need in the future (during and after the development of the software) for the manager to consult with the stakeholders about any change in the requirements of the application.

Engineering teams can use these methods to manage the complexity involved in creating smart, connected products. Using a requirements management solution helps to streamline the process so you can optimize your speed to market and expand your opportunities while improving quality. In order to be considered a "good" requirement, a requirement should have certain characteristics, which include being:

- ✓ Specific
- ✓ Testable
- ✓ Clear and concise
- ✓ Accurate
- ✓ Understandable
- ✓ Feasible and realistic
- ✓ Necessary

Sets of requirements should also be evaluated, consistent and no redundant.

Benefits of requirements management include:

- ✓ Lower cost of development across the lifecycle
- ✓ Fewer defects
- ✓ Minimized risk for safety-critical products
- ✓ Faster delivery
- ✓ Reusability
- ✓ Traceability
- ✓ Requirements being tied to test cases
- ✓ Global configuration management

Software design and modeling

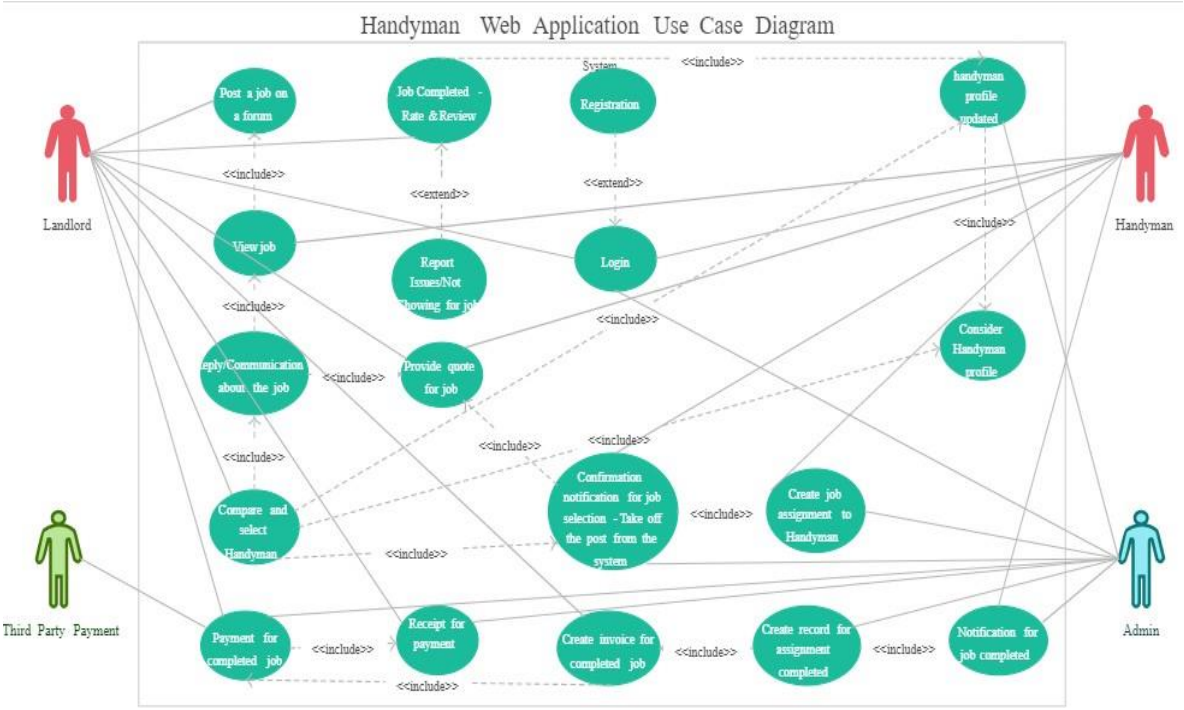
Through system design and modeling we are able to develop abstract models of the application where each new model being a fresh perspective of how the system works, is organized and interacts with other systems in some cases. Throughout this process we use graphical notations based in UML. As such, we as developers are able to understand and communicate more clearly every aspect of the application to the customer.

With the use case diagrams we display the interactions between the users and the system or other system. It includes actors that interact with use cases, use cases themselves which are the functions, the communication link between them and the system boundary to define what is inside and what is outside the system. Each use case can be described in more detail using a tabular description.

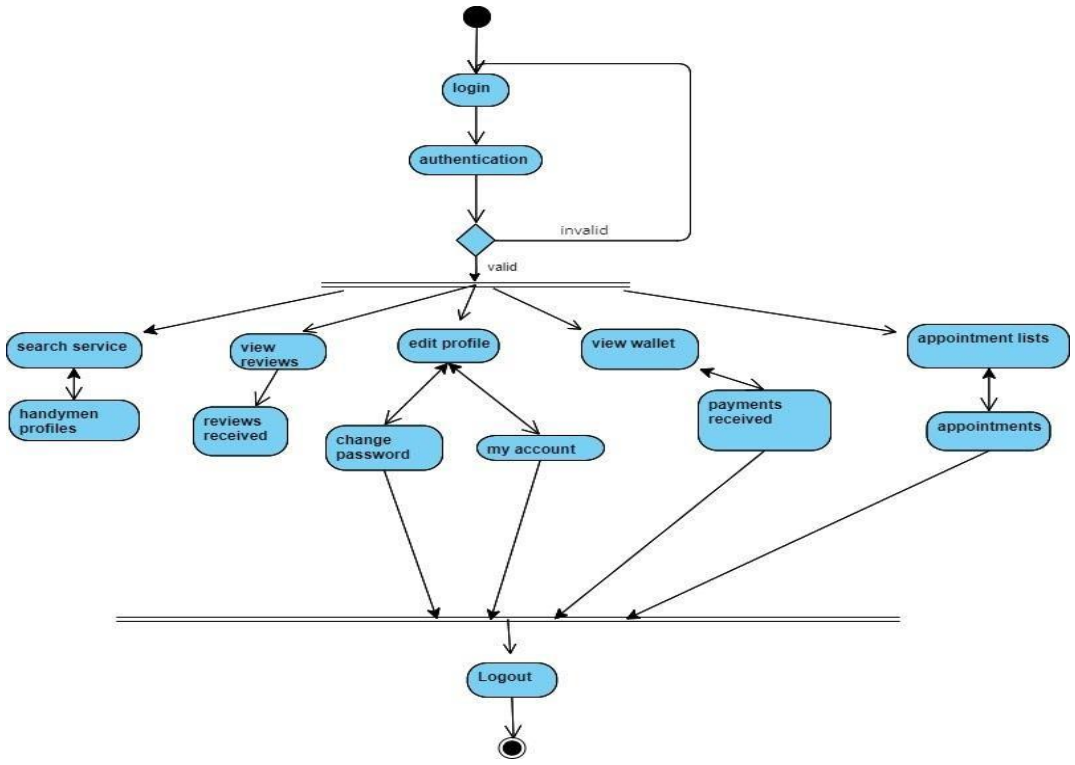
With sequence diagrams we show the sequence of interactions in the use case and the actors or objects are put at the top with a dotted line dropping vertically. The interaction between the objects is indicated by an arrow which shows a message and the dotted arrow a reply message, or synchronous and asynchronous. The vertical blocks from the lifelines or actors or objects indicate the time required, meanwhile in some cases we may have alternative blocks of what follows when an event occurs or not.

With the activity diagram we understand an advanced form of a flowchart where we understand the workflow. We start with an initial state and continue the action flow with activities sometimes branching and sometimes merging, forks or guards used at times too, as seen fit while ending with an end state.

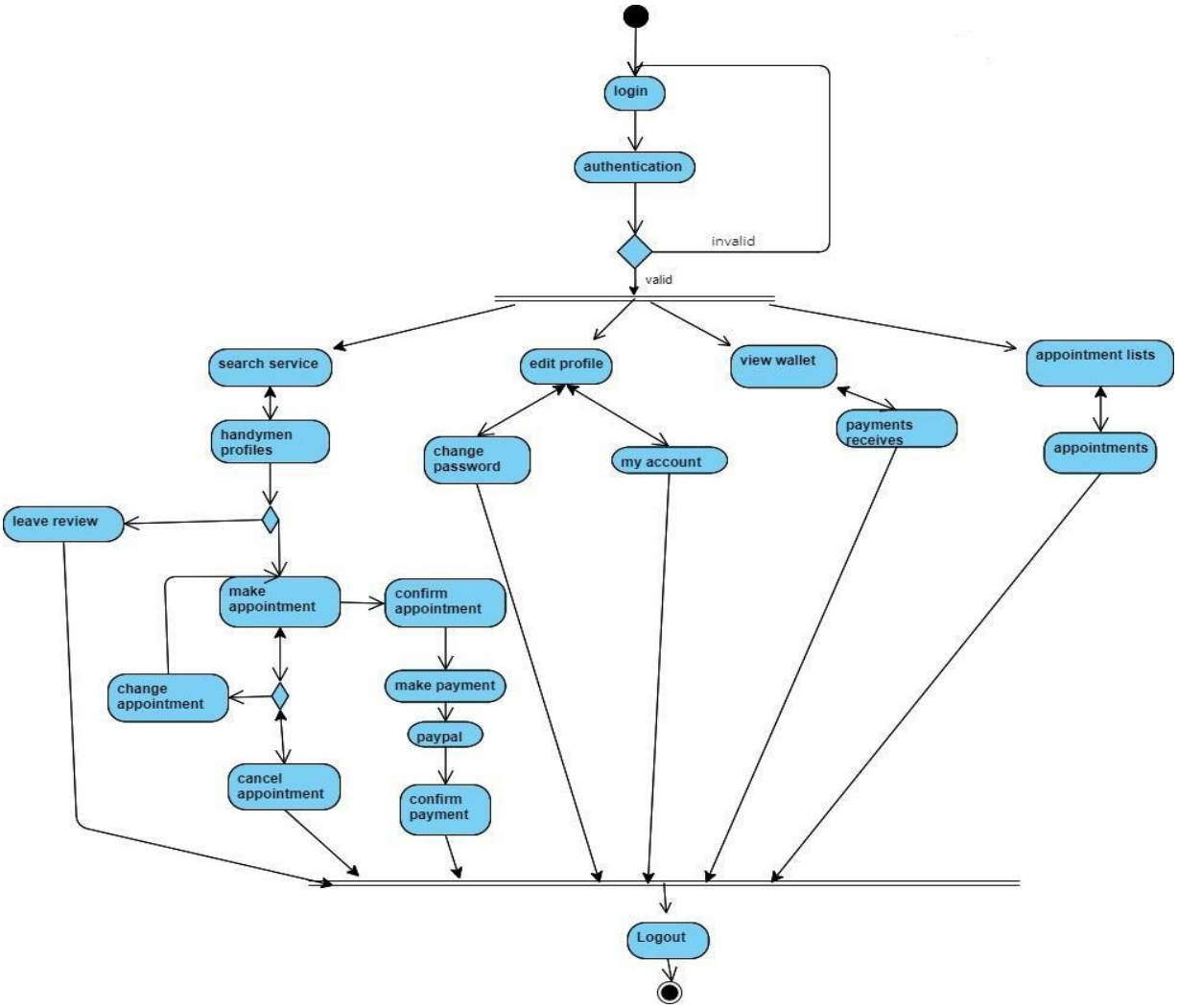
Handyman Application Full Application Use Case



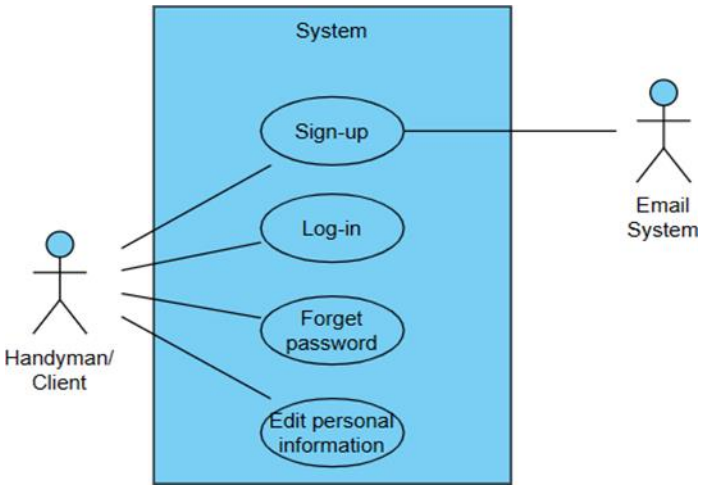
Activity model for handyman



Activity model for client



Use cases in the application regarding sign-up/login, forget password and editing personal information

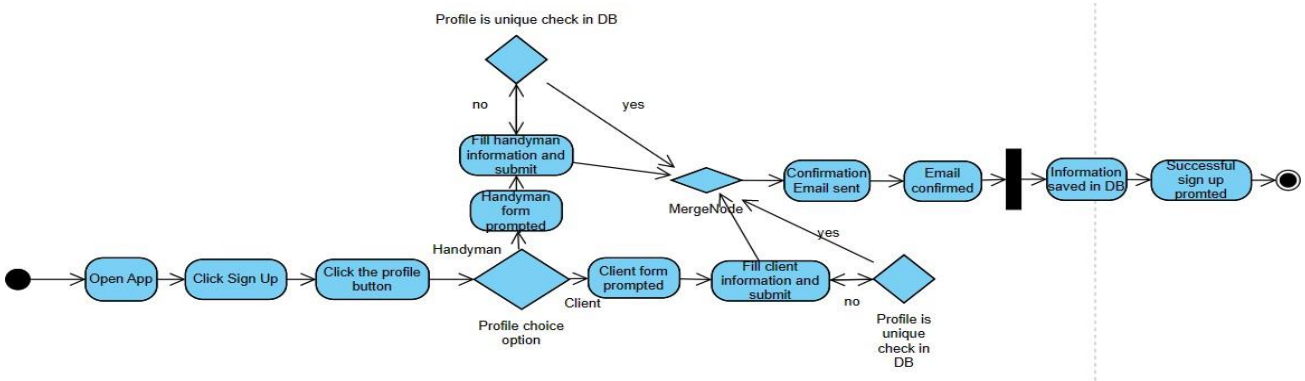


Sign-up use case

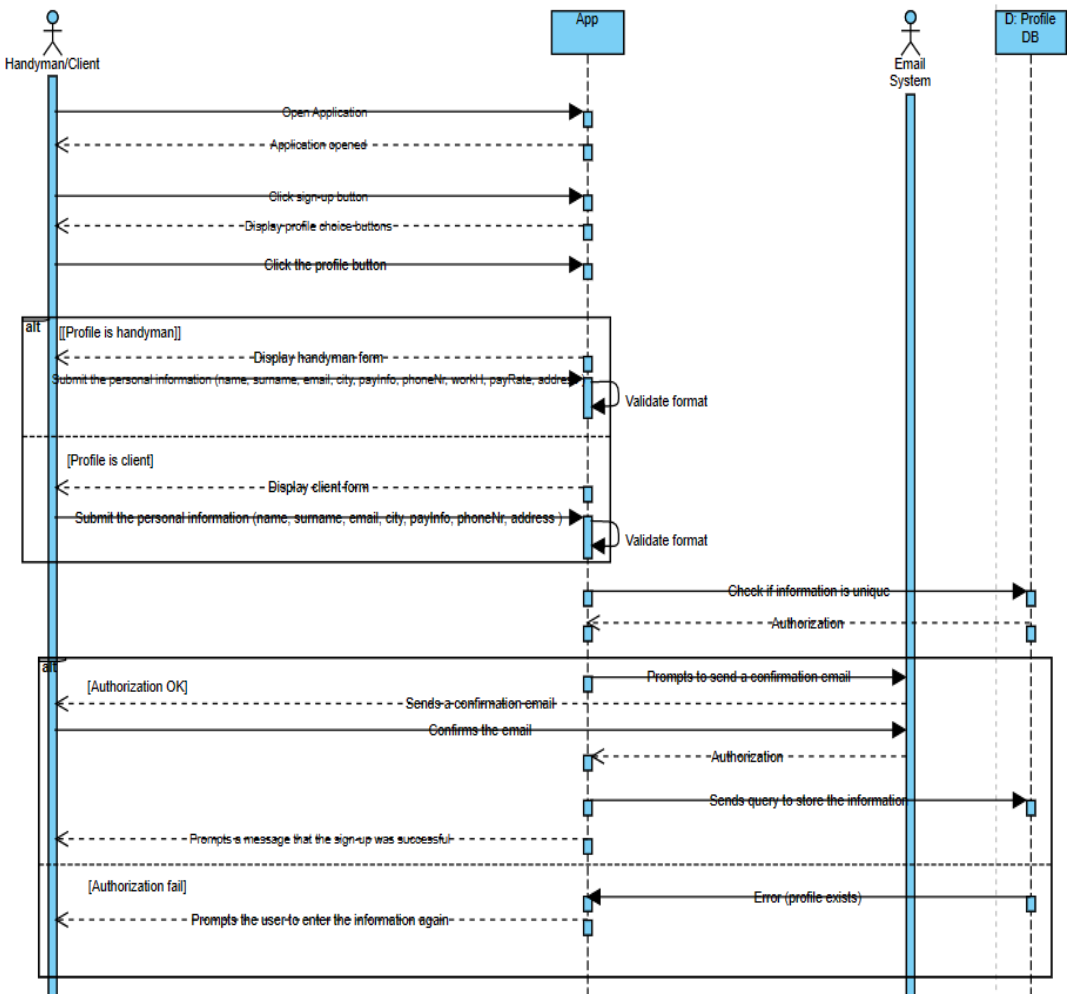
Tabular Description for sign-up:

Handyman Application: Sign-up	
Actors	Handyman/client, handyman application, email system, dB
Description	A user, be it a handyman or client, should be able to sign up with their personal information and create a profile or account in the application.
Data	User's personal information
Stimulus	User command issued by the handyman or client by clicking the sign up option
Response	Confirmation that sign-up was successful
Comments	It is key that the user specifies his/her role by clicking the profile of being either a handyman or client depending on their needs. The sign-up should be confirmed with a confirmation email from the application in the email specified by the user.

Activity Diagram for sign-up:



Sequence diagram for sign-up:

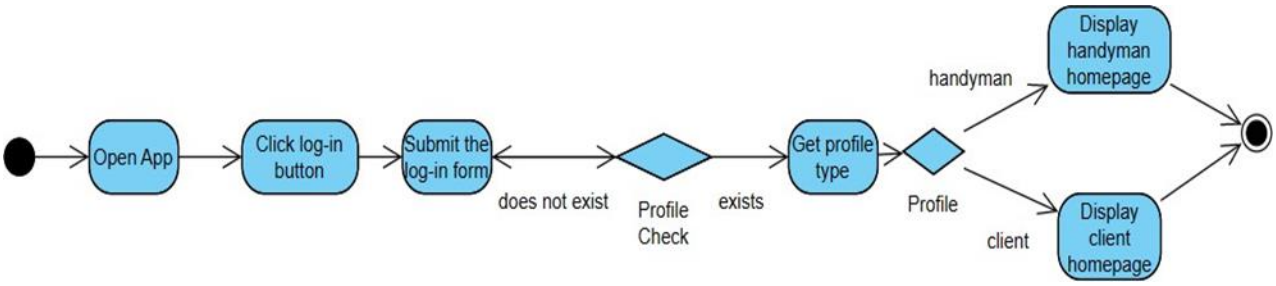


Log-in use case:

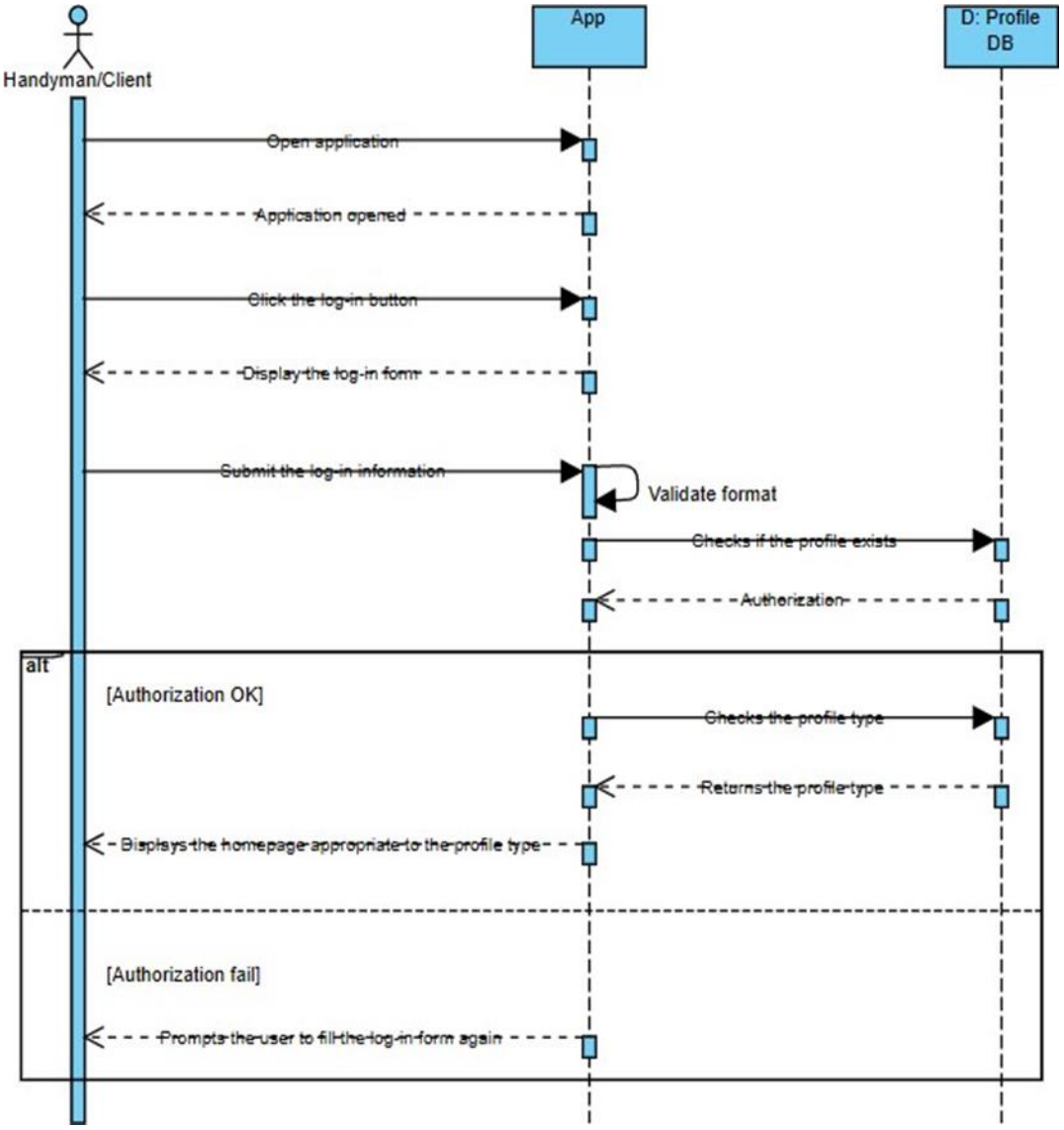
Tabular description for log-in:

Handyman Application: Log-in	
Actors	Handyman/client, handyman application, db
Description	A user, be it a handyman or client, should be able to log-in with their personal credentials and access the application.
Data	User's personal information
Stimulus	User command issued by the handyman or client by clicking the log-in option.
Response	Entrance to the application and being prompted the homepage.
Comments	Depending on the profile of the user depicted in the sign-up process, the user will be prompted a slightly different homepage with different options tailored to their profile needs.

Activity diagram for log-in:



Sequence diagram for log-in:

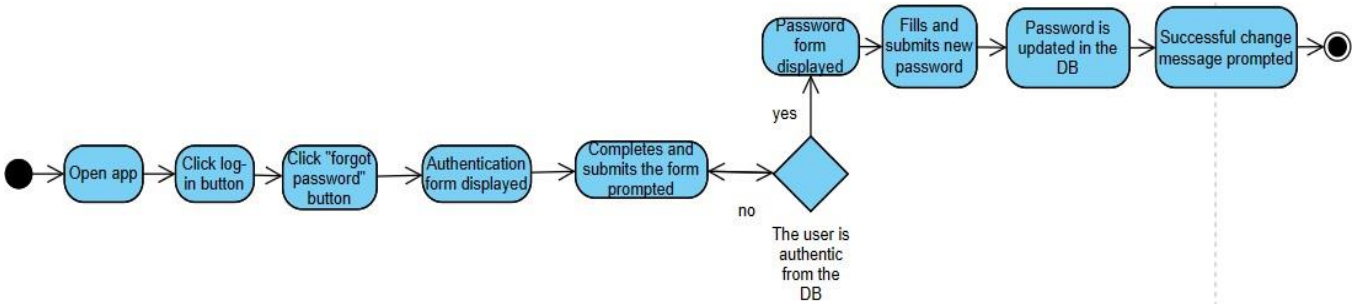


Forget password use case

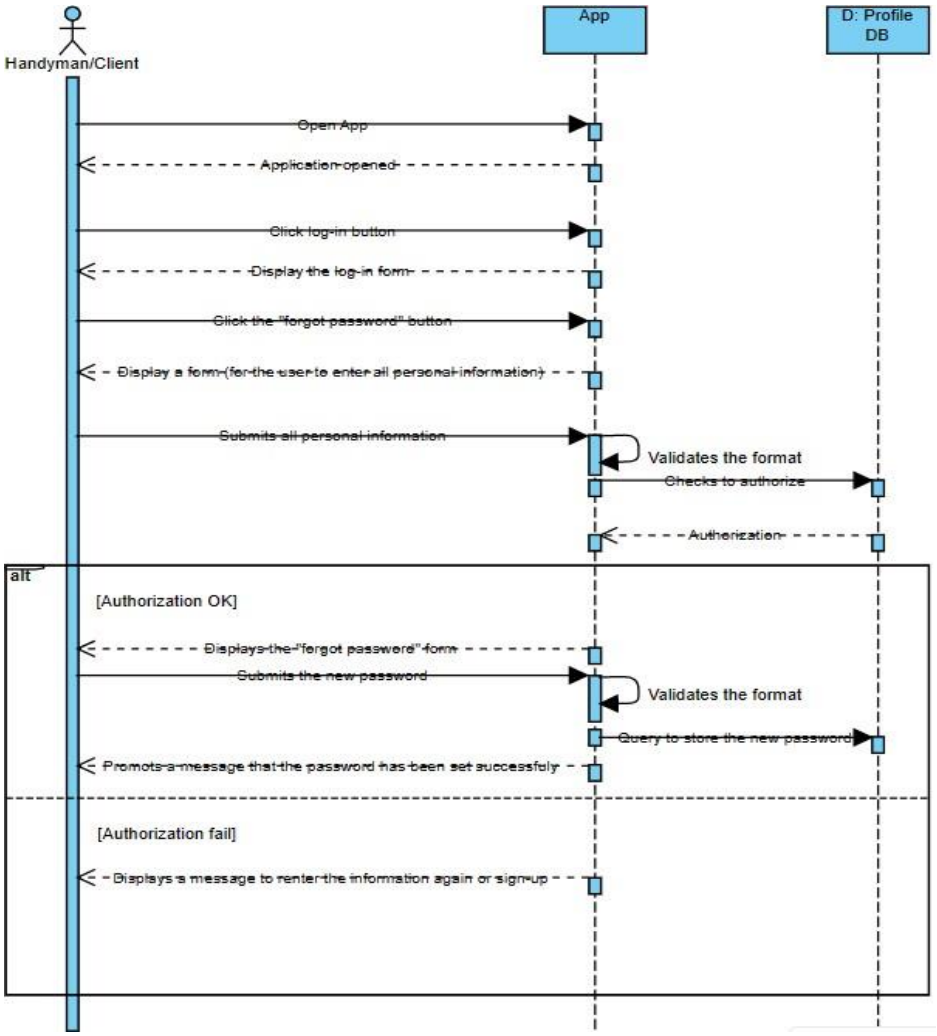
Tabular description for “forget password”.

Handyman Application: “Forget password”	
Actors	Handyman/client, handyman application, db
Description	A user, be it a handyman or client, should be able to reset their passwords if they have forgotten it.
Data	User’s personal information
Stimulus	User command issued by the handyman or client by clicking the “forget password” option.
Response	Confirmation the password has been reset successfully.
Comments	The user needs to remember all the other information that has been entered in the profile in order to be authenticated.

Activity diagram for “forget password”:



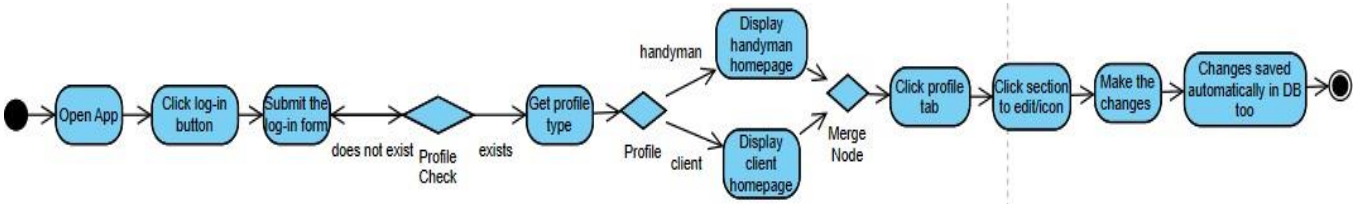
Sequence diagram for “forgot password”.



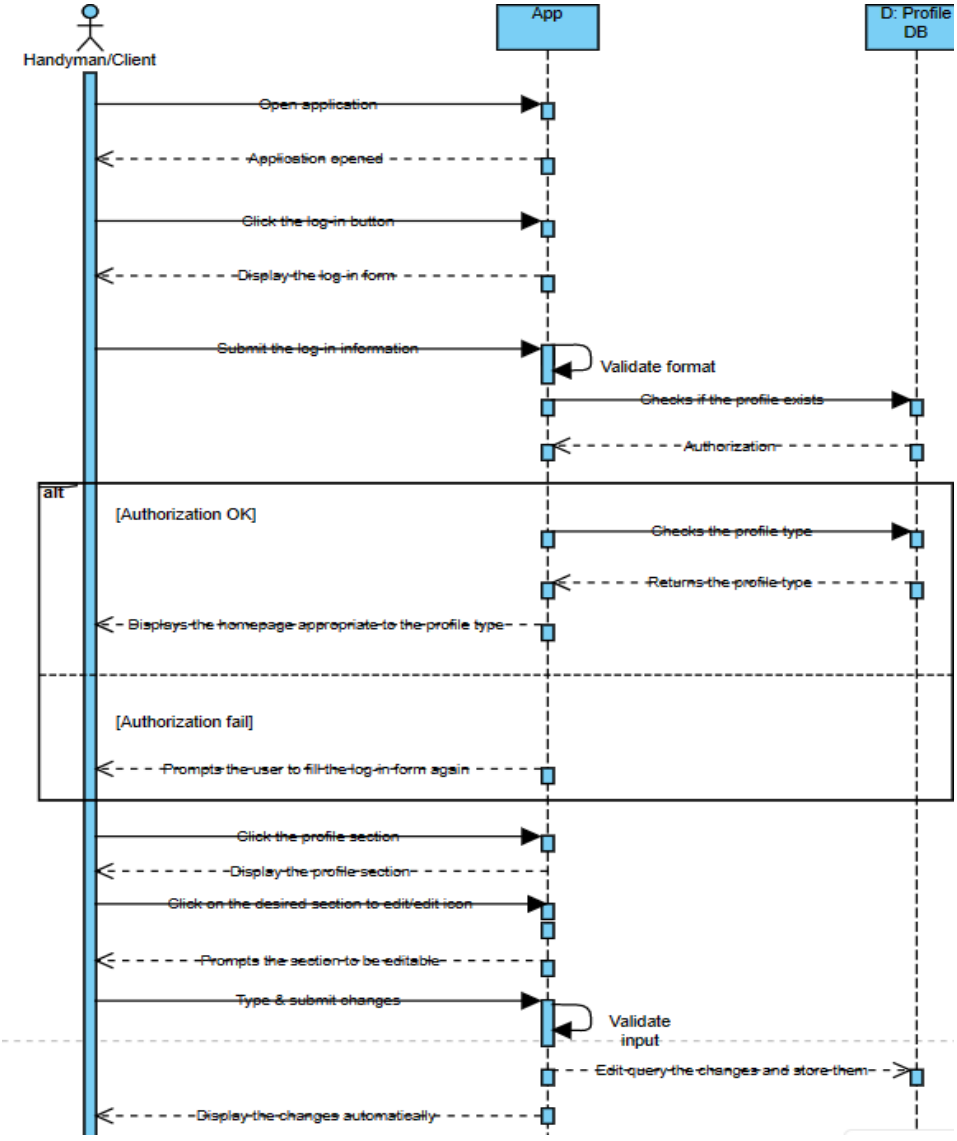
Edit personal information use case:

Tabular description for “edit personal information”.

Handyman Application: “edit personal information”	
Actors	Handyman/client, handyman application, dB
Description	A user, be it a handyman or client, should be able to edit their profile information from their profile section.
Data	User’s personal information
Stimulus	User command issued by the handyman or client by clicking the “profile section”, and then by either clicking the area they want to change or by clicking the pen/edit button.
Response	The edit will be reflected immediately as changed. No confirmation will be prompted.
Comments	This feature will allow the users to keep their profile up-to-date as needed.

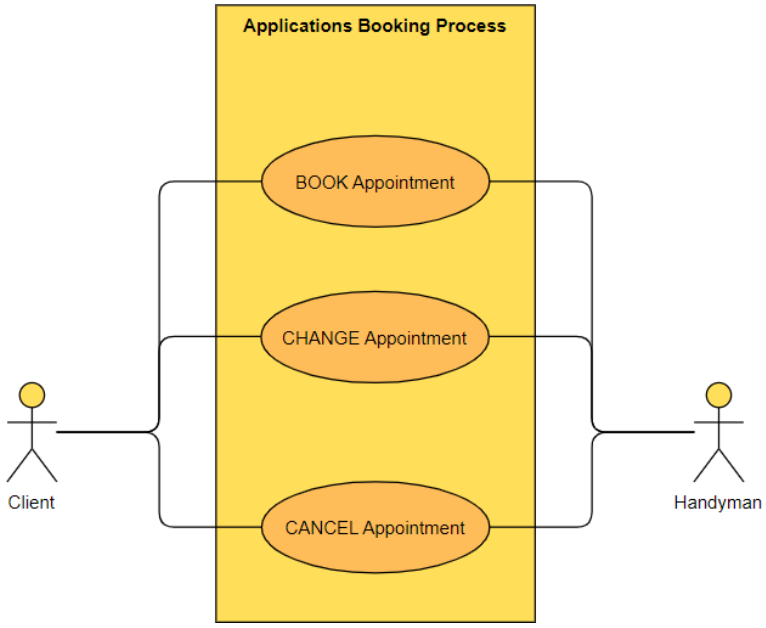


Sequence diagram for “edit personal information”:



Use cases in the application regarding Appointment

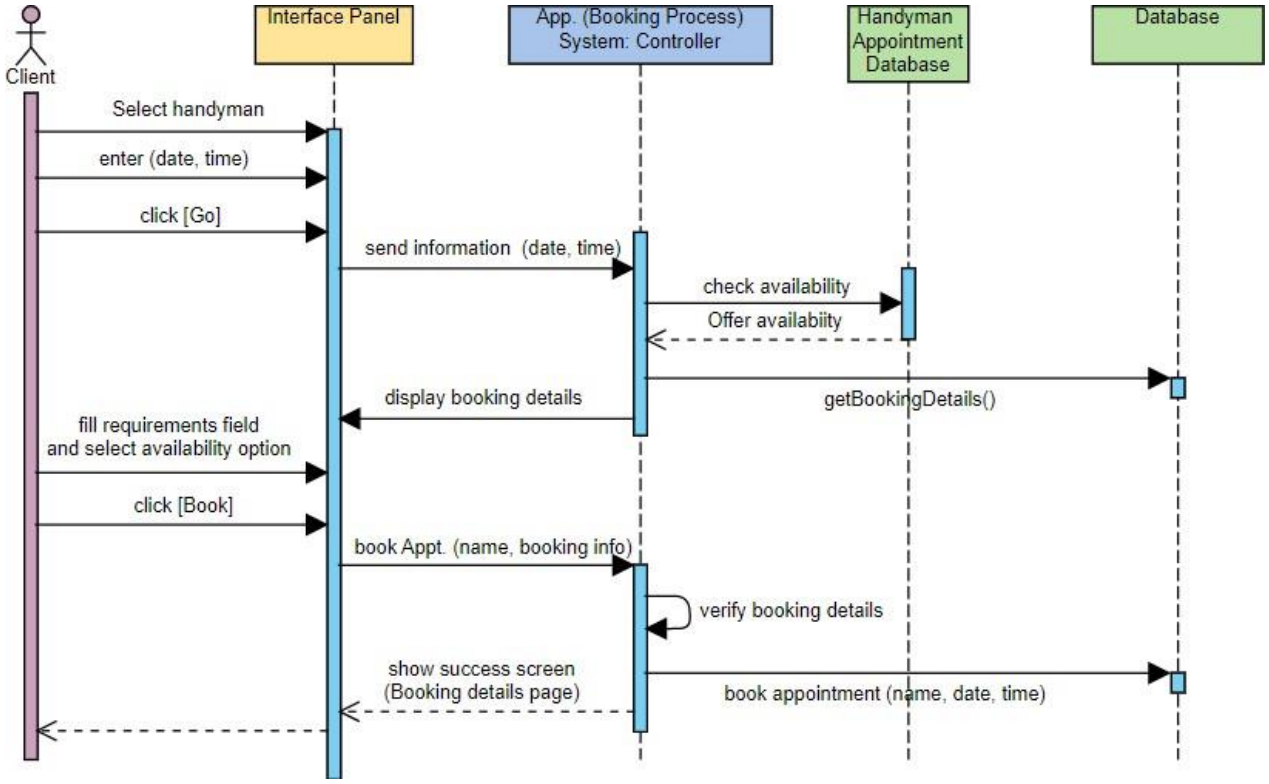
1. Book Appointment
2. Change Appointment
3. Cancel Appointment



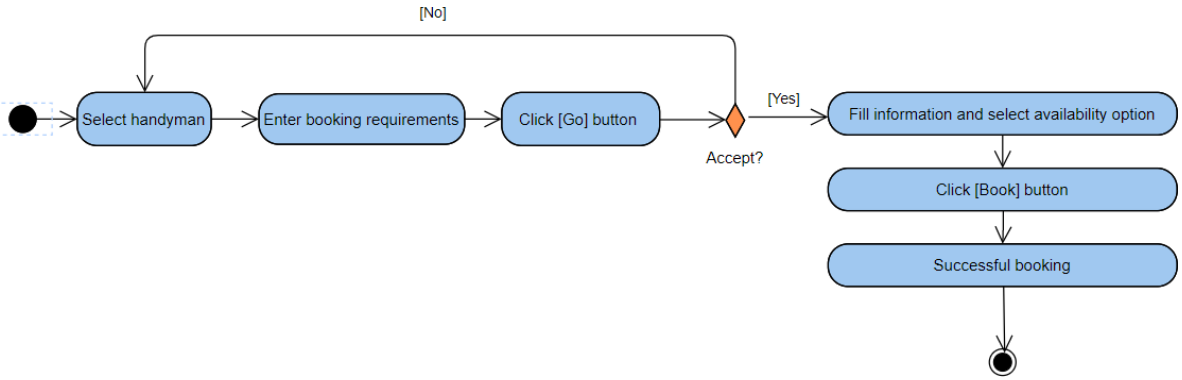
Tabular description for **Book Appointment**

Handyman Application: Book Appointment	
Actors	Client, handyman, handyman application, app.DB
Description	Users must be able to book, cancel or change appointments.
Data	Users' information, appointment/booking information (date, time)
Stimulus	User command issued by the client by clicking on "Book" button
Response	Confirmation of the booking procedure
Comments	User must specify his/her information correctly in order to check availability and confirm the service appointment by the service provider.

Sequence diagram for Book Appointment



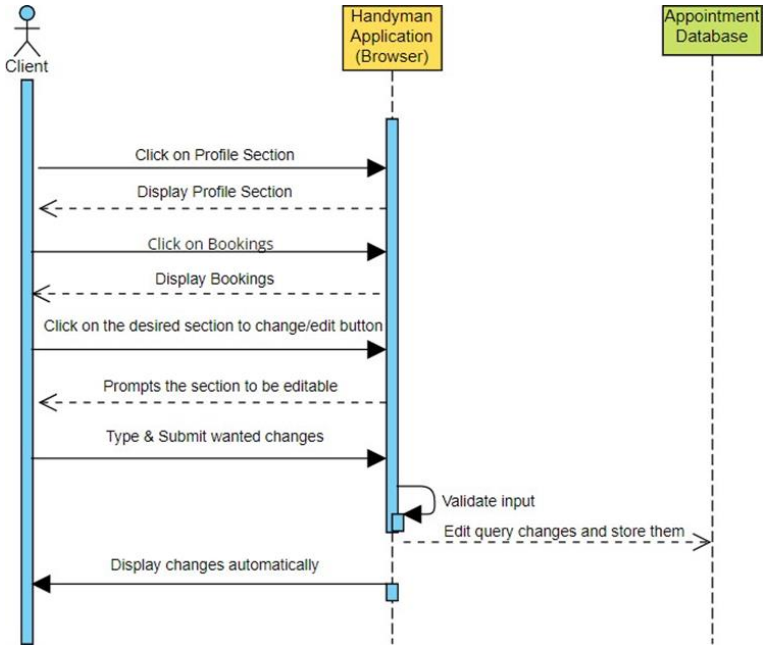
Activity diagram for Book Appointment



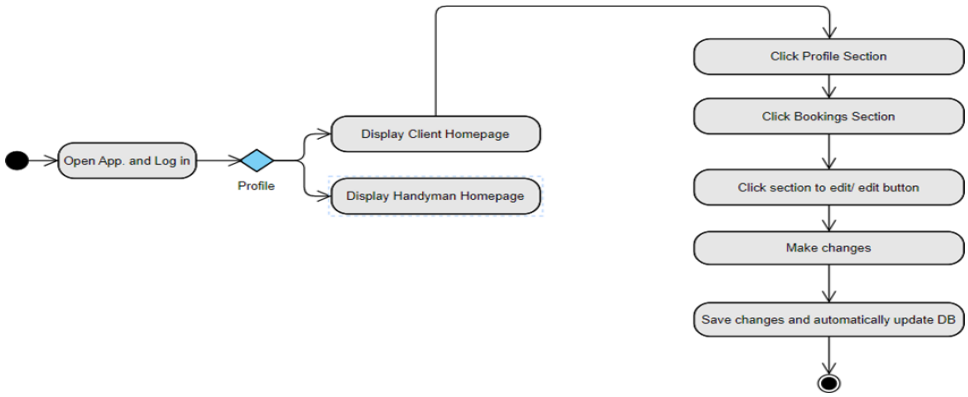
Tabular description for Change Appointment

Handyman Application: Change Appointment	
Actors	Client, handyman application, app.DB
Description	Users must be able to change their appointment in case there is a time conflict or anunforeseen change to the date or time set previously.
Data	Users' information, booking information (date, time)
Stimulus	User command issued by the client by clicking on "Change" button
Response	Confirmation of the changed appointment
Comments	User must specify his/her information correctly in order to check availability and getconfirmation for his booking.

Sequence diagram for Change Appointment



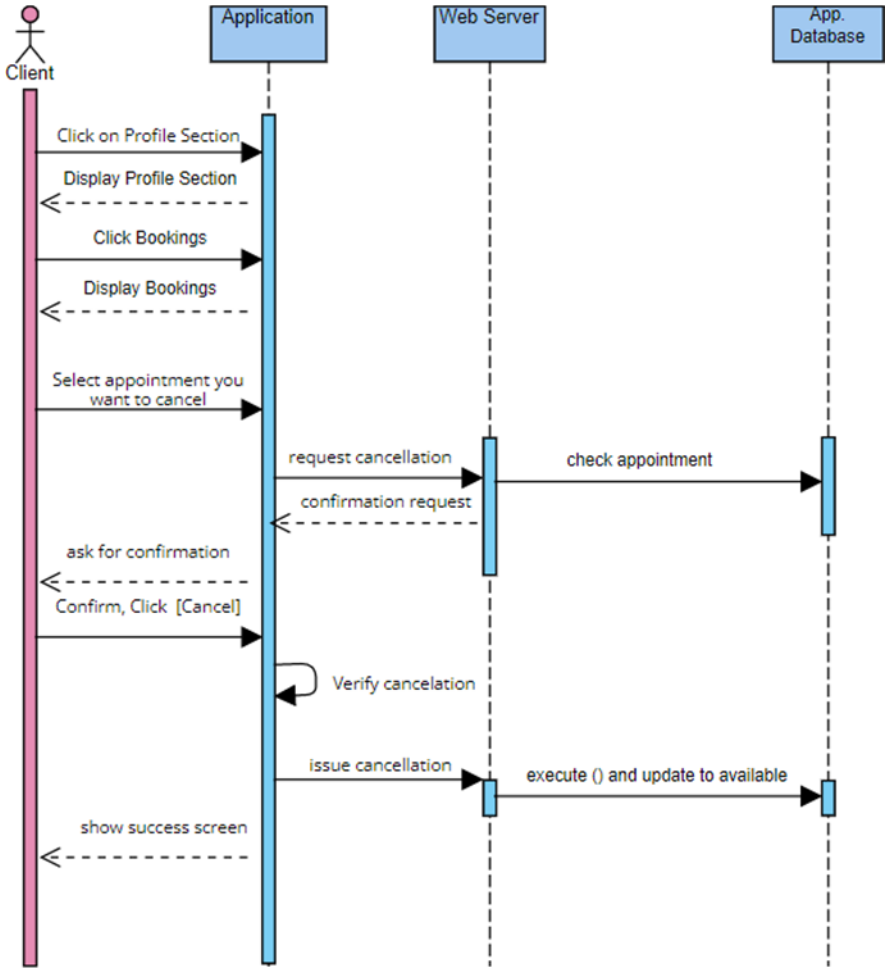
Activity diagram for Change Appointment



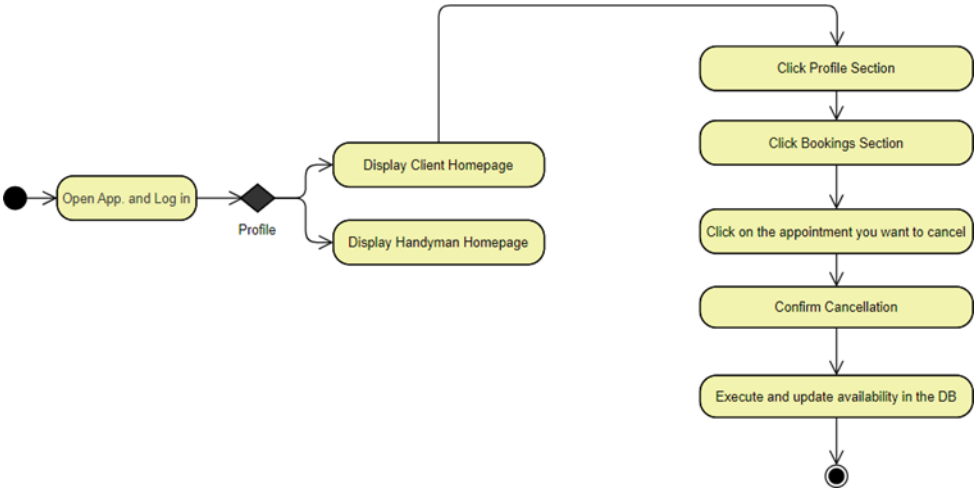
Tabular description for Cancel Appointment

Handyman Application: Cancel Appointment	
Actors	Client, handyman application, app.DB
Description	Users must be able to cancel their appointment anytime, at least 24 hours prior of their scheduled appointment, in case they no longer want to receive service.
Data	Users' information, booking information (date, time)
Stimulus	User command issued by the client by clicking on "Cancel" button
Response	Confirmation of the cancellation process
Comments	User must understand that at this moment his appointment has been cancelled and for this reason the requested service **will not be provided**. *User must be careful in case of confusion between *CANCEL* and *CHANGE* options.

Sequence diagram for Cancel Appointment



Activity diagram for Cancel Appointment

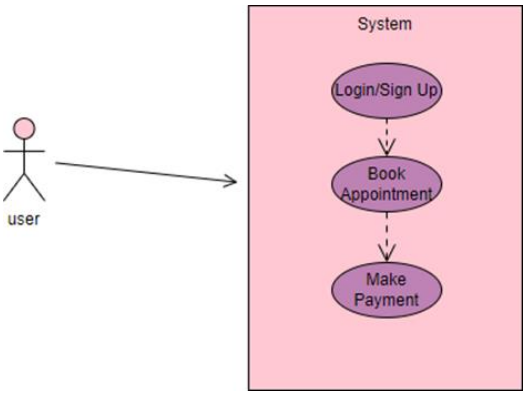


Explanation: In each sequential and activity diagram, the user is already logged in to the application and therefore has access to all functionalities the application provides. Since the login process is described thoroughly at the beginning of the designing and modeling section, the Booking System with its functionalities is explained separately as seen above.

Use cases in the application regarding Make Payment

Explanation: Taking into consideration that the sign up/login and booking appointment has been made, we proceed with the payment procedure.

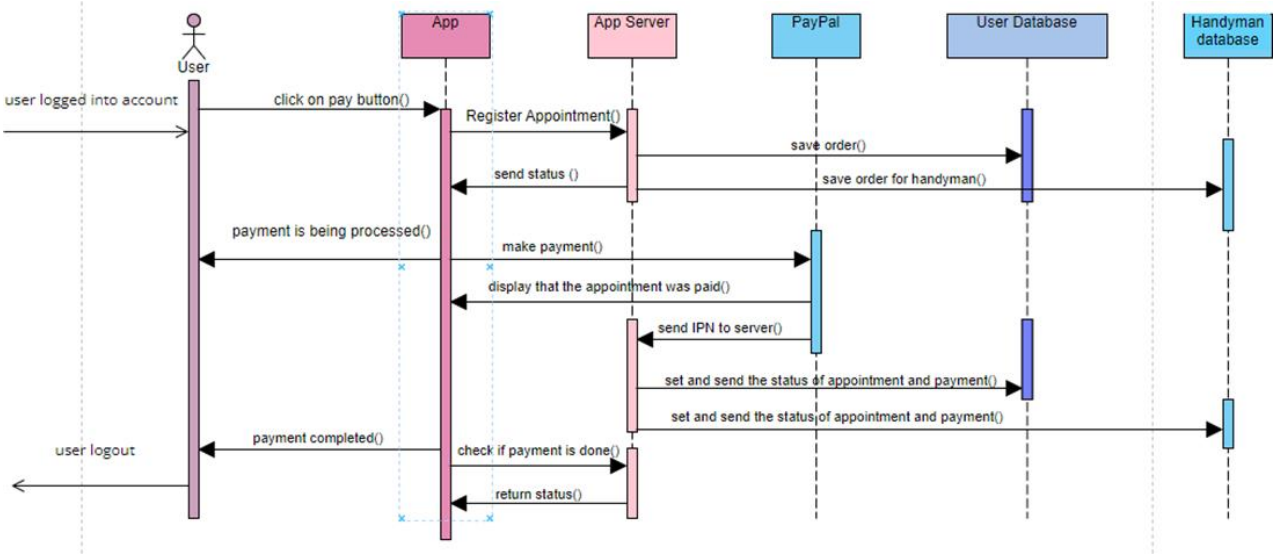
Use case for Make Payment



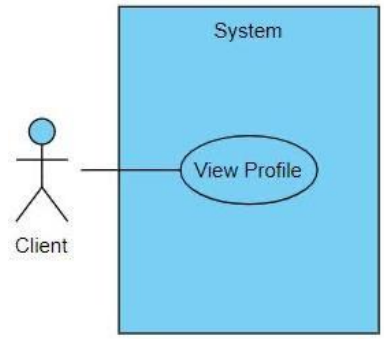
Tabular description for Make Payment

Handyman Application: Make payment	
Actors	User, handyman application, user DB, handyman DB
Description	User must be able to make PayPal payments.
Data	Users' information, booking information (date, time).
Stimulus	User command issued by the client by clicking on "Pay" button
Response	Confirmation of the payment procedure.
Comments	User must specify their personal information correctly in order for the payment to be done successfully.

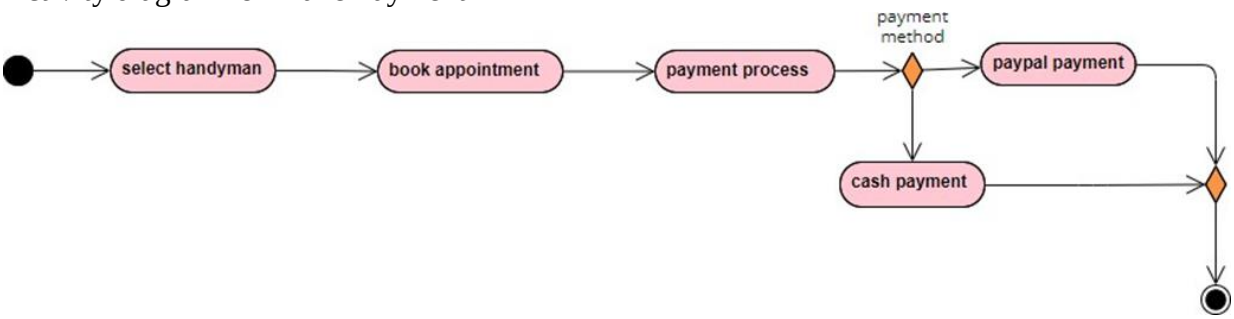
Sequence Diagram for Make payment



Use case in the application regarding view profile from the user.



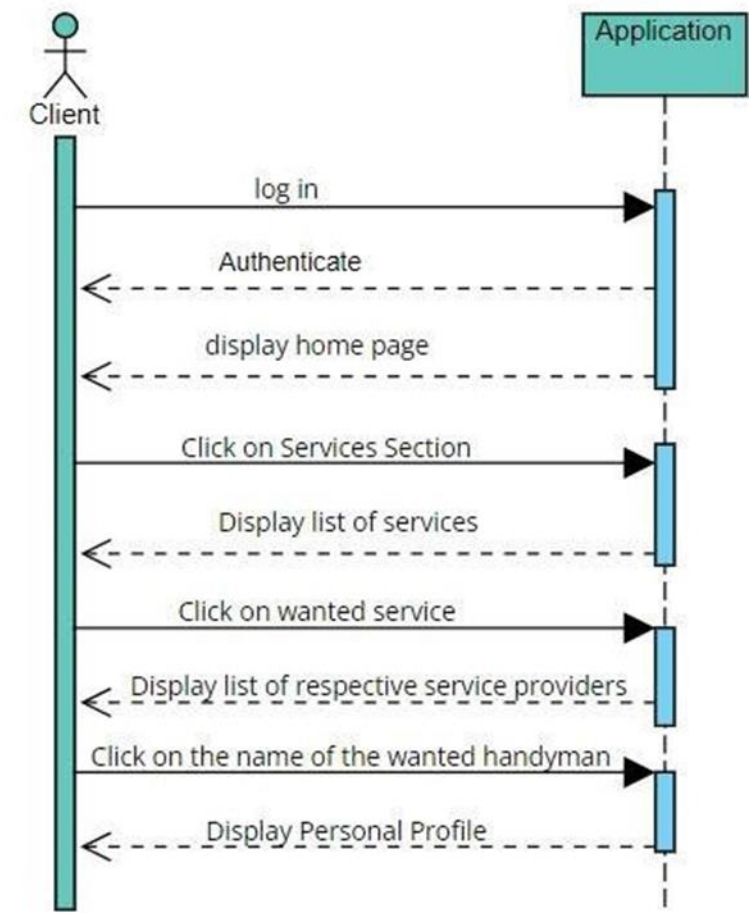
Activity diagram for Make Payment



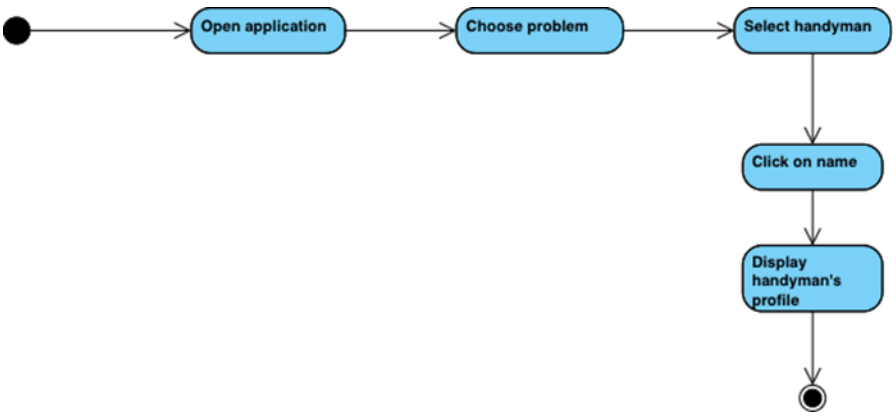
Tabular description of the "View Profile" use-case.

Handyman Application: View profile	
Actors	Client/The handyman.
Description	Both users are able to view other handyman's profile.
Data	Handyman's personal information (name, phone nr., city, home address, pay rate).
Stimulus	User command issued by clicking on "View Name".
Response	Display of Handyman information.
Comments	User must click on Handyman's name in order to generate this information.

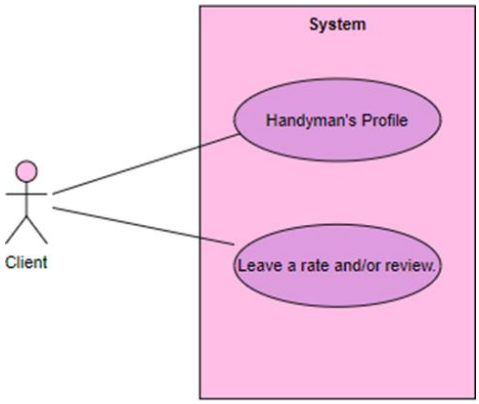
Sequence diagram for view profile



Activity diagram for view profile



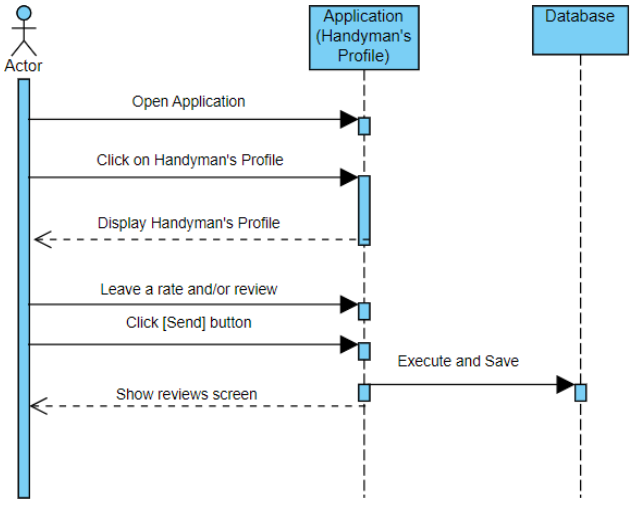
Use case in the application regarding Leave review from the user.



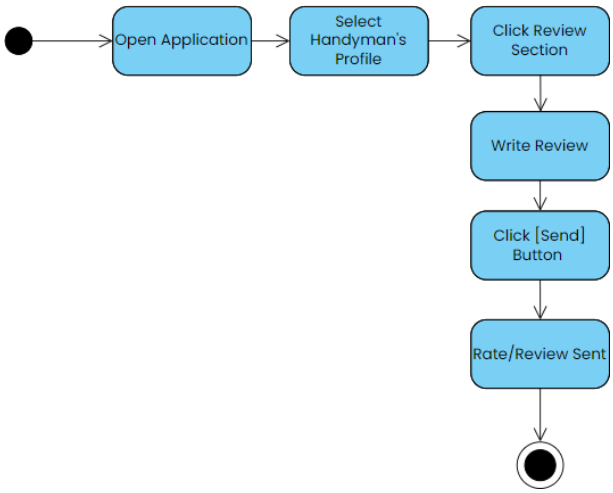
Tabular description of the “Leave Review” use-case.

Handyman Application: Leave Review	
Actors	Client.
Description	The client should be able to log in, rate and/or leave a review in handyman’s profile.
Data	User’s personal information.
Stimulus	User command issued by client by checking handyman’s profile and leaving a review.
Response	Confirmation that the review is posted in handyman’s profile.
Comments	The client should be able to leave rates and reviews in any handyman’s profile.

Sequence diagram for “leave review”



Activity diagram for “leave review”

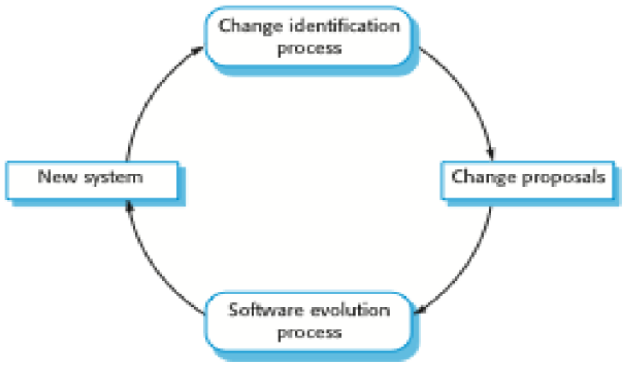


Evolution and Maintenance

Change identification and evolution processes

As we know from literature, evolution is "the stage in a software system's life cycle where it is in operational use and is evolving as new requirements are proposed and implemented in the system." Throughout this section we are going to delve deeper into this concept and perhaps take a bit of a hypothetical route since in real terms the software of this project is not being currently used in real life and in a real situation and business environment.

The change identification and evolution processes follow this cycle graph in the simplest terms containing 4 main stages:



1. Change identification process:

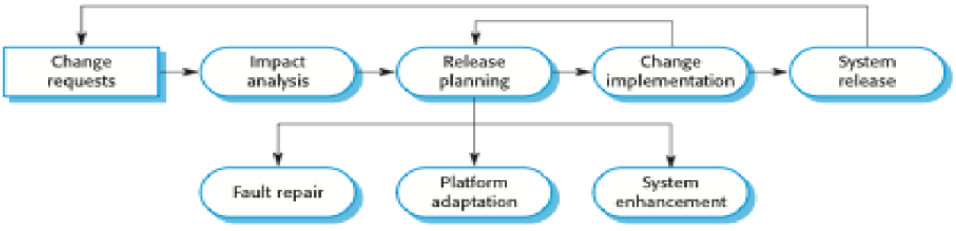
Let's say that the stakeholders come to the contracted company regarding the software that they have been using for some time and have found that they need to be able to update a technician's information because there have been times where they have entered it wrongly in the first place and could not change it.

2. Change proposals

A proposal for change is made where an update button is suggested to be added in the same row as the technician which would prompt an update form.

3. Software evolution process

This process in itself contains several processes such as:



1. **Change requests**

The request is to add edit functionality for the technicians.

2. **Impact analysis**

In short, we can say that this functionality will impact mainly the administration profile of the software; it will take about 3 business days to complete with a relative cost of $150. It will impact the administration profile which will allow the technician component to have the information editable and on the user side it will be the same with no change. It is a very isolated change that will only impact the code as an addition and not the architecture of the system for example or overall requirements and model.

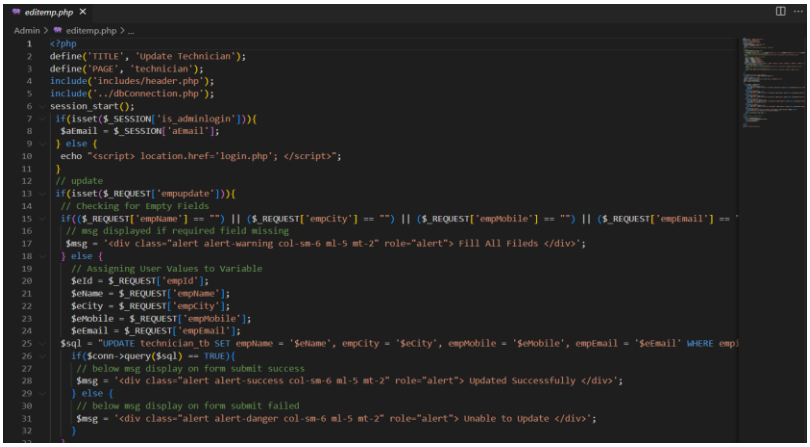
3. **Release planning**

In short, at this stage we plan firstly the release of this change by taking into account the fault repairing so what will be fixed, which is the technician component of the code, platform adaptation where it will not be necessary due to the nature of the addition and the same route would be followed and for system enhancement where we can say that this change will improve the system and it does not require or change any other functionalities or requirements.

4. **Change implementation**

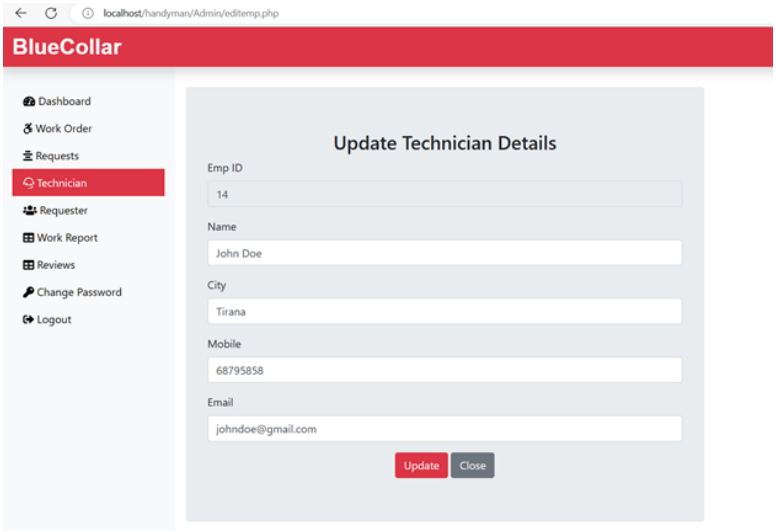
At change implementation we have another set of more detailed stages:

- The proposed change is for the software to allow the administrator to edit the personal information of a technician.
- The requirement is a functional requirement and its impact is mentioned above.
- Ideally the requirements document built in the beginning is updated with the new requirement.
- We continue onto developing the code that will implement the new requirement. At this stage we implement the change with the following code addition and the new file “editemp.php”. (This is just a small snippet picture and not the whole code, just for illustration purposes.)



5. **New system**

The system is delivered to the business or company that has contracted us with the new functionality in place. It is ideal that we keep in touch with them for any faulty instances.



The emergency repair process

If the business has a problem that is so urgent that it is disruptive to the businesses' operations resulting in loss of profit etc… then we follow a much shorter procedure as depicted below:

- ✓ First, we take the change request from the business to fix a certain bug.
- ✓ We analyze the source code to understand the root of the problem.
- ✓ We modify the source code and make sure that it performs as desired.
- ✓ We deliver the modified system to the client with utmost sense of urgency while still maintaining professionalism.

Handover problems

One handover problem we might face is that the evolution team might want to use an agile method, since we have used a plan-based one. It could be more difficult to them because they might be unfamiliar with the plan-based method and they would have to start from scratch developing automated tests and the code might not be refactored or simplified as expected in the agile method.

Bad smells

Coding errors or bad programming practices lead to code smells. These errors in the application code are frequently attributable to mistakes the application programmer made when writing it. Code smells are typically the result of standards not being followed when writing the code.

Examples of bad smells are:

- ***Duplicate code:*** where the same or similar code may be included in different places in a program.
- ***Data clumping:*** This occurs when the same group of data items re-occurs in several places in a program.
- ***Switch case statements:*** Include duplication. In our project, we have used different codes in different parts of the program. No group of the same data re-occurs in several places. Each code is different and there are no duplications or data clumping.
- ***Long methods:*** If a method is too long it should be redesigned as a shorter method. In our code, there are no long methods including too many different topics, but they focus on a single thing.
- ***Speculative generality***: This occurs when developers include generality in a program in case it is required in the future. We have been specific regarding the program we have created and no generality has been used.

Issues in business value assessment [4]

The main goals when building an application are the creation of an application that serves a certain function and solves issues in a cost-effective manner for the respective actors. The original software must be modified and changed due to the business needs and the market's numerous advancements and changes; thus, it is crucial to conduct some preliminary research in order to decide on the best practices to follow throughout this period. In order to conduct an accurate assessment, we should take different perspectives into account such as system end-users, business customers, line managers, IT managers, and senior managers. As we said previously, as software evolves there are often multiple features and enhancements that could be

implemented. Business value assessment helps in prioritizing these based on their potential impact on achieving business goals and objectives. It makes sure that all the allocated resources are used for the most valuable cases that will lead in the future to improve efficiency and maximization of performance.

Some of the issues regarding business value assessment are as follows:

1. The use of the system: Given that our application is the hub of all "BlueCollar" business activity, its use is crucial for the upkeep and running of that firm. Because of this, a significant percentage of users regularly utilize this program, demonstrating its strong business value.
2. The business processes that are supported: This application's structure avoids requiring its users to use ineffective business processes, hence facilitating the work of all stakeholders who use it. In consideration of this, it is safe to claim that it has a high business value.
3. System dependability: As of right now, we can say that the program is reliable. Problems are rare and typically not directly tied to the customer of the business. For the moment this seems as a positive indicator and value.
4. System outputs: As we already mentioned, “BlueCollar” business activities are all conducted through the utilization of the application. Therefore, we can say that the business depends on system outputs indicating that the system has again a high business value.

System quality assessment

By guaranteeing the continued quality of the software system, system quality assessment plays a crucial role in software evolution. To ensure the efficacy and dependability of software as it develops and changes, it is necessary to evaluate and maintain its quality.

Business process assessment

Assuming that the business we are working on is a medium-sized business, we can say that so far, all business processes support the current goals of the business.

As we have explained in the other sections, the application is divided into two distinct interfaces, each of which has the functionalities required to enable business processes for the users who are using it. In addition to their unique functionalities, there are simple processes between them that make the connection between the client and the company possible without posing challenges or difficulties for any party.

Environment assessment

Environment assessment is another standpoint that we must take into consideration when analyzing system quality. By evaluating the technical, operational, and organizational

environments in which the software runs, environment evaluation plays a significant role in the software evolution phase. It helps ensure that the evolving software is compatible, scalable, and resilient. When evaluating and assessing the environment some factors must be taken into account, such as failure rate, performance, age of hardware and software components, and maintenance cost. Regarding "BlueCollar" we can say that the technological infrastructure is new and considerably good. Therefore, until now the failure rate is low and the performance of the system is adequate. Standard practice in the software industry is to budget about 15-20% of the original cost of developing the app per year for maintenance. Regarding our application, only 15% of the cost of the application will be set aside for annual upkeep.

Application assessment

Application assessment is another important perspective that focuses on evaluating the current state of the software application and its components. In order to effectively drive software evolution, it aids businesses in understanding the application's strengths, limitations, and potential improvement areas. Again, there are some factors that we must consider when conducting the assessment of the application. Some of the factors used are understandability, documentation, performance, programming language, and personal skills. We can say that the source code is relatively easy to be understood and the control structures used are not complex. Alongside there are offered the respective documentation in order to easily understand the application such as system and requirements, process, planning, standards, technical and architecture & design documentation. The documentation is complete and consistent. Regarding performance, we can say that it is adequate and has not caused any major problems for system users. Overall, the program is running smoothly in all directions. Undoubtedly, personnel skills also play an important role in this direction. The team is prepared and has all the necessary skills required to maintain the application. In conclusion, we can say that the application is running smoothly but there is always room for evolution and development in order to offer the best possible service.

Maintenance Prediction

Maintenance prediction is the process of predicting when maintenance tasks will need to be performed on a website in order to maintain its ideal functioning, security, and performance. It seeks to foresee faults or future maintenance requirements and proactively identify them before they result in major downtime or delays. By anticipating maintenance needs, website administrators can schedule and organize maintenance tasks in advance, minimizing the impact on users and the possibility of unforeseen issues.

In order to make a prediction we need certain data and information related to the website that we are building like:

1. ***Data collection and analysis:*** Since we have built this website from scratch we do not have any historical data to look at, however we do have data and information of the current website so keeping track of them, documenting and analyzing them as often as possible will impact the maintenance prediction easier and more accurate.
2. ***Metrics classification:*** We must first determine the metrics on which this forecast will be based in order to determine whether the website requires maintenance. Website traffic, security flaws, user reviews, page load times, server response times, and database performance are a few measures.
3. ***Continuous monitoring:*** In order to determine in real time whether our website requires assistance or not, we must remain constantly aware of its activity. Because it is impossible to avoid change, maintenance must be performed on our website constantly.
4. ***Predictive models:*** Developing models that can forecast future maintenance needs based on previous data and recognized patterns using a variety of methodologies, including statistical analysis, machine learning, or predictive analytics. These models can help understand what type of maintenance should be done, when and in which part of the code.

Predicting maintainability

What parts of the system will be the most expensive to maintain?

Since this website is a Custom-built one which is developed from scratch, it can be more costly in generally to maintain. Usually codes that are built specifically for that certain business require updates, bug fixes, sometimes even have unique features. The most expensive part of our system i believe would be protecting all the data. Since it requires sensitive data to book, cancel, change an appointment, create an account and most importunately pay it is crucial for us to invest highly in making sure that all this data is protected when shared and stored.

Predicting maintenance costs

What will be the lifetime maintenance costs of this system? What will be the costs of maintaining this system over the next year?

Cost forecasting is challenging since it is reliant on several situations and variables. This cost will never be zero, as technology is continually developing. The following elements must be taken into consideration in order for us to anticipate costs:

- *Updating the server:* Websites often require regular upgrades in order to solve any issues that may arise over time; these frequent changes may have an impact on cost.
- *Correction of mistakes and bugs:* this step must be completed. They do pay for a significant percentage of the expense since conception and execution are created specifically for each customer, eliminating the possibility of errors.
- *Tech support:* it may include troubleshooting, debugging, solving technical difficulties. Troubleshooting, debugging, resolving technological difficulties, and user support are all examples of tech support.
- *Security:* The cost of deploying security measures like firewalls increases due to the requirement of routine audit, vulnerability assessments, and protection against possible attacks.
- *Redesign:* A website may need to update its brand in order to react to shifting business demands and fashion trends.

It is significant to highlight that because to the numerous factors involved, correctly calculating lifetime maintenance costs are difficult. In addition, unforeseen problems and ongoing technical improvements may have an impact on maintenance requirements and related costs. As for lifetime maintenance I cannot make any predictions as far as numbers go but I believe it would go between the lines of $200 to $400 early.

Predicting system changes

What part of the system is most likely to be affected by change requests?

Usually the parts of the system that are predicted to change more frequently are profile of users and handymen and appointments. We have left it in their hands to change personal information as many times as they need even passwords and usernames. Also they are free to change, cancel and book appointments as many times as they like, it will be updated accordingly to the database.

How many change requests can be expected?

Since we have a waterfall method, it's expected to have some requests from our clients. Even though we have done an amazing job at the documentation process, still it exists a possibility for the clients to have different requests at the end of the website design.

Maintenance Cost Factors

Maintenance cost factors play a crucial role in the development of software. One significant factor is team stability, which has a direct impact on the overall quality of the software and helps to reduce errors. We as a software development team have maintained consistency from start to finish by getting a thorough understanding of the software's architecture, codebase, and functionality. The familiarity allows us to resolve any concerns quickly while maintaining a high level of quality.

Once the final product is delivered to the client, our team assumes contractual responsibility for its maintenance. This responsibility creates an incentive for us to design the software in a way that facilitates future changes. By considering the long-term maintenance aspect during the development phase, we can implement modular and scalable solutions. This approach makes it easier to update, improve, and adapt to changing user requirements. We can reduce future maintenance costs and ensure the software's adaptability and robustness by being proactive in our design choices.

Another vital aspect related to maintenance cost factors is the skill set of the team members. In our case, our team possesses the necessary skills and expertise required for efficient software maintenance. We have a diverse range of technical competencies, encompassing various programming languages, software development methodologies, and debugging techniques. This diverse skill set ensures that we are well-prepared to handle different maintenance tasks effectively, such as bug fixes, performance optimizations, security updates, and feature enhancements. Our collective knowledge and experience enable us to troubleshoot issues promptly, minimize downtime, and provide timely support to our clients.

Process Metrics

Process metrics play a crucial role in assessing the maintainability of a software system. These metrics provide insights into the efficiency and effectiveness of the maintenance process. There are some process metrics that can be used to evaluate maintainability:

Number of Corrective Maintenance Requests: This metric tracks the frequency with which requests are made to fix flaws or address issues in the software system. An increase in corrective maintenance requests indicates an increased likelihood of system flaws or problems, potentially indicating a loss in maintainability, and it signals that the program may require more time and resources to keep it functioning properly.

Average Time Required for Impact Analysis: Impact analysis refers to the assessment of the potential effects that a proposed change may have on the software system. This metric measures the average time taken to analyze the impact of a change request. An increasing average time for impact analysis may indicate a decline in maintainability and it suggests that the system's complexity or lack of proper documentation is hindering the ability to understand the potential consequences of changes.

Average Time to Implement a Change Request: This metric calculates the average time it takes to implement a requested change in the software system. If the average time to implement a change request increases over time, it may signal diminishing maintainability and that the system's architecture or codebase is getting more difficult to adapt, maybe due to tight coupling or insufficient flexibility.

Outstanding Change Requests: This metric monitors the number of change requests that have been filed but have not yet been implemented or handled. An increasing number of pending change requests could indicate a loss in maintainability, implying that the system's capacity to manage change requests is being overwhelmed, possibly due to resource constraints or poor communication among stakeholders.

Monitoring these process metrics allows us and stakeholders to identify potential issues and take corrective actions to keep or improve the maintainability of the software system. By addressing emerging trends or bottlenecks in these metrics, we can improve our capacity to manage changes effectively, reduce errors, and assure the software's durability.

Task Allocation

Phase 1

Alesia Gjana: *Problem description and solution*

Gloria Muskaj: *Aim and main objectives*

Aldi Hamati: *Description of the application*

Phase 2

Erjola Avdiaj, Amanda Gaci: *User requirements and application specifications*

Gloria Muskaj, Aldi Hamati, Alesia Gjana, Amanda Gaci, Erjola Avdiaj: *Software design and modeling:*

Gloria Muskaj, Aldi Hamati, Alesia Gjana, Amanda Gaci, Erjola Avdiaj: *Software Development*

Gloria Muskaj, Aldi Hamati, Alesia Gjana: *Maintenance plan and Software Evolution*

Phase 3

Aldi Hamati: *Introduction, use cases in the application regarding sign-up/login, forget password and editing personal information*

Gloria Muskaj: *Use cases in the application regarding Make Payment, activity model for user and handyman*

Alesia Gjana*: Use cases in the application regarding Appointment, use case of application*

Amanda Gaci: *Use case in the application regarding View Profile*

Erjola Avdiaj: *Use case in the application regarding Leave review*

Phase 4

Aldi Hamati: *Change identification and evolution process, change emergency process*

Gloria Muskaj: *Maintenance Prediction*

Alesia Gjana: *Issues in business value assessment*

Amanda Gaci: *Maintenace cost factors*

Erjola Avdiaj: *Handover problem*

References

[1] https://www.indeed.com/career-advice/career-development/common-functional-and-non-functional-requirements

[2] https://www.digitalguardian.com/blog/what-pci-compliance

[3] https://termly.io/resources/articles/legal-requirements-for-websites/

[4] https://iversoft.ca/what-does-it-cost-to-maintain-an-app/