Team Name: Blue Collar
Team Leader: Gloria Muskaj
Team Members: Gloria Muskaj, Aldi Hamati, Alesia Gjana, Amanda Gaci, Erjola Avdiaj

Roles and Tasks

Phase 1

Problem description and solution: Alesia Gjana
Aim and main objectives: Gloria Muskaj
Description of the application: Aldi Hamati

Phase 2

Requirements Elicitation and Analysis: Aldi Hamati
User and System Requirements, Domain Requirements, Model Development: Gloria Muskaj
Functional and Non-functional and Domain Requirements: Alesia Gjana
Requirements Validation: Amanda Gaci
Requirements Management: Erjola Avdiaj

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 to mine 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 cluttered.
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.

   1.1 The first page of the application should have the login and sign up button.

   1.2 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.

   1.3 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.

   1.4 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.

   1.5 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.

   2.1 The moment that the user opens the application (after logging in), they will be greeted by the main/home page.

   2.2 The home page would hold a difference between the taskers and the clients. The clients home page will include about us, services section, meanwhile the taskers home page will include about us, appointment and wallet section (text area).

   2.2 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.

   2.3 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.

   3.1 After logging in, creating an account the user will be able to pay for every service.

   3.2 The user will be asked for the transaction requirements after they have filled the required fields in the booking form.

3.2 Create an HTML form to collect the customer's payment source information.

3.3 Create a token to transmit the encrypted data securely.

3.4 The next step is to submit the form with all relevant data to the customer's source bank for the requisite approval.

3.5 Once the customer's source bank confirms the transaction, verify and validate the charges involved in the order process.

3.6 Whether the transaction is approved or rejected by the customer's source bank, display the transaction's status to the consumer.

3.7 The payment details need to be stored in the database.

3.8 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.

4.1 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.

4.2 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.

4.3 The handyman should be able to change their profile information as many times as they want (dynamic system).

4.4 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.

4.4 Every change that the tasker does will automatically be updated to the database.

4.5 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.

5.1 Every handyman should have in their profile a text area where users will be able to leave their reviews and ratings as they prefer.

5.2 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.

5.3 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.

6.1 The user must be able to book an appointment with the handyman.

6.2 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.

6.3 When the handyman is booked they should have an updated HTML list of their booking time and day in their profile.

6.4 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.

6.5 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.

6.6 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.

7.1 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.

7.2 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.

7.3 If moving further when clicking the name of a certain service provider the button will redirect to their profile page.

7.4 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

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*	MUST	

user's data.		
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* homepage where 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 behaviour 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

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***
  - 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:

- ✓ 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.

- ✓ 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.

- ✓ Define and record: We create a document where all the requirements the stakeholders asked for are listed

- ✓ 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

- ✓ Agree on and approve requirements: After agreeing on the requirements listed, we start to implement them in the application

- ✓ 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