Fudee

Nihu Rao
CS7316-701C-1227
nbalabommala@smu.edu

Sushma Chiluvuri
CS7316-701C-1227
schiluvuri@smu.edu

Abstract

This paper studies and identifies two major problems of the commuters in the United States. It mainly focuses on the Fudee application that has been developed till prototype stage. Further, it studies three Requirement elicitation process methods that have been performed to gather and enhance the requirements of the Fudee application. Finalized requirements are presented along with the prototype that is explained in detail. It then studies forward requirement traceability that has been performed on the application.

Keywords: Commuters, fuel delivery, carpooling, requirement elicitation, collaboration, brainstorming, questionnaire, traceability.

1 Introduction

Commuting alone by car has become more common in most countries. [1] According to Statista's survey, United States have highest number of commuters who travel by their own car to work/university when compared to other highly populated countries like China, Germany, United Kingdom, South Korea, and Brazil. This survey included 1,600 to 6,200 participants.

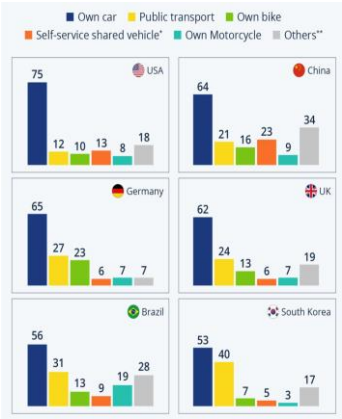

Figure 1 Commuters around the world

In above figure, we can see that only the United States have more than 75 percent of commuters who prefer travel by their own car whereas China has 64 percent of commuters, Germany has 65 percent of commuters, UK has 62 percent of commuters, Brazil has 56 percent of commuters and South Korea has 53 percent of commuters.

To study in detail about the majority of Americans who use their cars to travel to and from work/school, a study has been examined. [2] According to Statista's Global Consumer Survey, the percentage of Americans who commute by their own car to work was 76 percent. Making owning a personal vehicle the most common means of mobility. And among the 5,649 respondents, just 11 percent use public transit.

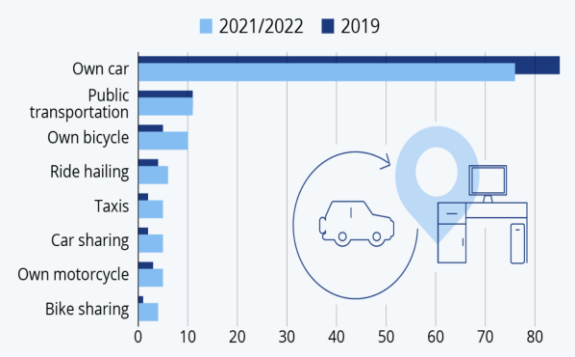

Figure 2 U.S residents who commute to work

The Figure 2 shows the survey data results that have been recorded in the year 2019 with 3,338 respondents and 2021/2022 with 5,649 respondents. In the year 2019, there were 85 percent of respondents preferred to use their car compared to other ways of transportation. In the year 2021/2022, the percentage of respondents who use own car for commuting has dropped from 85 percentage to 75 percent due to COVID-19 pandemic. However, even in the pandemic most commuters travel by their own car compared to other ways of transportation.

Traveling by own car can be the most expensive way of travel due to increasing fuel prices. [3] According to U.S. Energy Information Administration, the gases prices have been drastically increasing over the years.

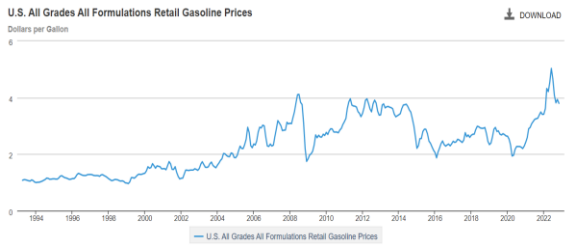

Figure 3 US gasoline prices over the years(monthly)

From the above figure, we can see that the first drastic increase from the year 2004 to 2008 and the second major drastic gasoline price increase is from the year 2020 to 2022.

One of the major problems caused during commuting to work or university is emergency situations like running out of fuel. This emergency can occur anywhere, when travelling in a city or while traveling to and from isolated places. When this situation occurs, there are chances where the gas stations are far from the commuter location or there are no gas stations in some deserted places.

Two major problems have been identified. Firstly, traveling alone by car is an expensive way of travel. Secondly, during emergency situations such as running out of fuel while commuting to work/university or in deserted places. A proposed solution for these problems has been explained in detail in the below sections.

1.1 Proposed solution

Based on the problem identified we have a proposed solution. A platform can be developed to let the users access the platform to avail the services that enable them to share their trip plans and destinations, as well as services that can supply fuel when users post their whereabouts. And we need a system that enables users to speak with one other about their trip plans, discuss their destinations, and share their whereabouts so that gasoline can be delivered. Along with which the solution should be able to be user-friendly, and portable and should provide user rewards when service is utilized.

With this solution, the users do not need to stand in lengthy queues to refill their car fuel tanks and can get instant access to gas anywhere by simply pressing a button at any time.

1.2 Implementation of the solution

An application is developed to provide two services, carpooling and fuel delivery services. The application was developed using Android Studios and MySQL Workbench. Languages used for this application are Java, SQL, and PHP.

2 Requirement Elicitation

[4] Identification of needs is the process of eliciting requirements. The front end of systems development encompasses social, communication, and technological difficulties. It is helpful to express the demands systematically. The requirements analysis procedure comes after this one.

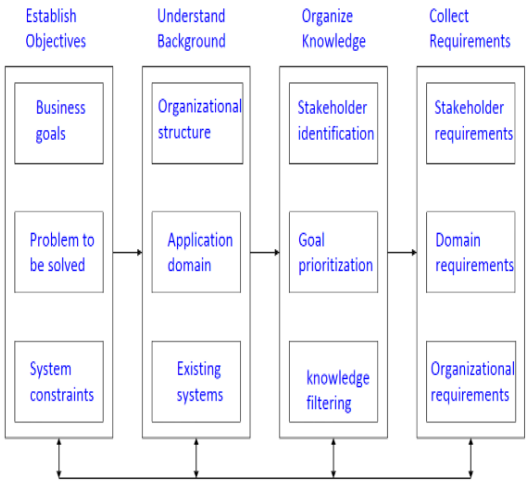

Figure 4 Requirement elicitation

The fundamental point of need elicitation is shown in figure 4. These pursuits were combined with one another. The general requirement elicitation process involves establishing objectives, understanding background of the organization, organizing knowledge and collecting requirements.

Three elicitation strategies were employed in this project to collect needs. These are

1. Brainstorming technique
2. Questionnaire technique
3. Collaboration technique

The ideal approach to employ each technique has been researched, meetings have been held, conversations regarding our app idea have taken place, and the needs have been obtained.

2.1 Brainstorming

Brainstorming is a useful method for coming up with original ideas for solutions to a specific situation. Each person in a brainstorming session is free to generate a huge number of ideas, which may be random, non-random, or even irrelevant. The key is to include every participant to produce the most ideas possible. Members are encouraged to contribute their thoughts and ideas during a brainstorming session without regard to the quantity or quality of those ideas.

We used a SWOT analysis to collect specifications for our app. An application session with seven stakeholders that lasted 25 minutes was held in order to implement this method. This conversation covered Fudee's advantages, disadvantages, possibilities, and dangers.

The app needs to be portable for all mobile devices and constantly accessible for users, according to all seven stakeholders. They liked how we shared a login and registration process and informed them of the travel time.

Requirements ID	Description
NFRS-001	The App shall work on all devices of android, iPhone 4S, and upward models.
NFRS-002	The app must be always available for users.
FRS-006	The App shall supply the users with the chance to register, log in, and mark their status as "I can share a ride" and set their destination on the map before starting the journey.
FRS-002	The App shall add up travel time duration of all user's present locations, journey distances, and final destinations.

Table 1 Requirements gathered

The above table is the requirement gathered using the brainstorming technique.

2.2 Questionnaire

In our project, using this strategy we gathered information from a sample all at once and a set of questions were given to our stakeholders to gauge their opinions. We discussed improvements in our application flow and how the interface feels like also a few ideas regarding the reward system. After gathering their replies, the data is analyzed to determine the stakeholder's area of interest.

In the final portion of the Session Phase, the team requested that the stakeholders fill out the questionnaire form that was used to collect useful information to improve the application. Through the questionnaire form, the team hoped to elicit the pros and cons of the Session Phase, areas for improvement, the experience of the stakeholders during the meeting, and most importantly, how well the phase was able to elicit adequate security requirements.

Set of questions were prepared and presented them in writing to ten stakeholders along with our app idea and flow of application (in sketch). On the questionnaire form, these questions were asked:

1. Do you think that our app idea needs improvements?
2. What do you think about the application flow?
3. What does the interface feel like?
4. What do you think about our reward system idea?
5. Does our reward system need any improvement?

Ideas: Responses have been gathered from each of the 10 stakeholders. They have acknowledged that when a user requests gasoline delivery, the program logs the user's precise position. They have advocated for the incentive system to incorporate gift cards, electronic currencies, etc. in addition to vouchers. They proposed that we hold a monthly fortunate draw to entice additional users. They proposed that the software should have a straightforward user experience for the third question.

Requirement ID	Description
FRS-003	The app shall record the exact location of the user when requesting fuel delivery.
FRS-004	The app shall supply a reward system for the users. Instead of a reward like vouchers, gift cards, or electronic coins, etc., and there will be a monthly lucky draw reward system conducted for the users.
NFRS-003	The App shall have a simple intuitive user interface.

Table 2 Requirements gathered

The above table is the requirement gathered using the questionnaire technique.

2.3 Collaboration

The key component of collaboration technique is stakeholder collaboration. Stakeholders with various viewpoints, who are frequently system users, actively participate in gathering, developing, or fine-tuning requirements throughout this co-creation process.

These methods provide people a forum for conversation and enable immediate input from interested parties. Workshops on requirements, crowd-based requirements engineering, and living laboratories are a few examples of collaboration methodologies.

A 30-minute meeting with six stakeholders (users) has been held to use this strategy. Delivering the app's fundamental idea, anticipated features, and flow were all part of this discussion. based on the type of comments received.

The app's stakeholders have made several suggestions to enhance its functionality, including one that would add a new requirement.

Ideas: Stakeholders were given a presentation of our application idea, and during the discussion that followed, they advised that in addition to the rider's current position, the search for rides also include the rider's intended destination.

The interested parties should be given contact information for ridesharing and fueling, according to the stakeholders, as a simple method of communication and for safety.

Requirement ID	Description
FRS-001	While supplying the ride posts, the app must search for rides based on the destination supplied and according to the rider's location.
NFRS-004	The app shall facilitate contact information between interested parties for sharing ride and fuel filling.

Table 3 Requirements gathered

The above table is the requirement gathered using the collaboration technique.

3 Prototype

Figure 5 Fudee app icon

The above figure is the app icon for the Fudee application that has been developed based on the proposed solution. Once the icon is clicked, the user is forwarded to registration and login page.

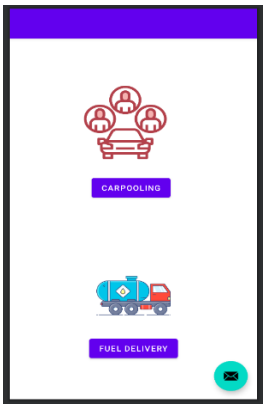

Figure 6 Home page of Fudee

The above figure is the home screen of the application. The user is forwarded to the homepage where the user has the option to either take a ride or post a ride by providing the necessary information. The user will be taken to the gasoline delivery homepage after selecting this button, where they must provide information about their car and choose the kind of fuel they want delivered. The user must be able to find rides when looking for ride posts by using both the destination and the user's location as criteria. The user can also get rewards based on the reward system which we implemented in our application according to the ride history of the user.

In this application, when the carpooling button is clicked the user will be redirected to next page where the user must choose between post a ride or take a ride. The rides will be posted in rides page. And when the option take a ride, the rides are searched based on the user location and the destination where the user want to travel to. And the available rides are provided. The total time duration is calculated by summing up the user's present location, journey distances, and final destinations.

For fuel delivery service, the fuel-delivery button has to clicked in the home page which redirects to another page. Here, the user must opt for sharing live location of the user and have to provide details like fuel type, car type, quantity of the fuel to be delivered.

At the end of both the services, payment must be done by the user. Along with utility of the services, the user is provided rewards with the rewarding system.

Requirement ID	Description
NFRS-001	The app shall work on all android devices.
NFRS-002	The app must be always available to users.
NFRS-003	The app shall have a simple intuitive to user interface.
NFRS-004	The app shall facilitate contact information between interested parties for sharing ride and fuel filling.

Table 5 Non-functional requirements

4 Finalized requirements

The descriptions of the system services and restrictions that are produced during the requirements engineering process are the requirements themselves**.**

4.1 Functional requirements

[5] Functional requirements are statements outlining the functions the system should do, how it should respond to certain inputs, and how it should operate in specific circumstances. the system's improper behavior might be stated.

Requirement ID	Description
FRS-001	While suppling the ride posts, the app must search for rides based on the destination supplied and according to the rider's location.
FRS-002	The app shall add up the travel time duration of all the users present location, journey distances, and final destinations.
FRS-003	The app shall record the accurate location of the user when requesting fuel delivery.
FRS-004	The app shall provide rewarding system for the users which include vouchers, gift cards, electronic coins, etc and there will be monthly lucky draw conducted for the users.
FRS-005	The app must charge 50% of the total ride to the user who has cancelled the ride.
FRS-006	The app shall provide the users to register, login and to mark their status as "I can share a ride" and set its destination on the map before starting the journey.
FRS-007	The app shall supply rewards based on a specific number of rides taken or rides provided, and the first-time user gets rewarded with a reward.
FRS-008	The app shall provide the options to select the fuel type, car type while utilizing fuel delivery app.

Table 4 Functional requirements

4.2 Non-functional requirements

[5] Restrictions on the products or services the system offers, such as deadlines, limitations on the development cycle, standards, etc. frequently refer to the system as opposed to specific features or services. These specify the system's characteristics and limitations, such as its dependability, responsiveness, maintainability, scalability, portability, and storage needs.

5 Requirement traceability

Intensive iterative procedure entails identifying and persistently labeling each need so that they may be referred to during the development cycle. software systems' increasing complexity and the ongoing requirement for swift evolution and update. A way to prove compliance with a contract, specification, or rule is through the following requirements. Enhance product quality while lowering maintenance costs and facilitating reuse. Maintain a balance between stakeholder demands and system development. knowing the software that is being developed and its artifacts. Effective change management skills. Ensuring that the environment in which the product operates and the product itself are both consistent. the capacity to specify, capture, and follow the footprints that requirements leave behind on other software development process elements as well as the footprints that those elements leave behind on requirements. Tracking requirements' linkages and dependencies is made easier with the use of traceability links. monitoring the change's spread once a specific requirement is removed or changed.

Defining every necessary relationship. Determining the product components for which traceability data should be kept. Selecting the traceability matrix type to be used. Outlining the tagging guidelines used to categorically label each requirement. Name the important people who will provide each type of link information. Bringing the team up to speed on the principles and value of requirements tracing. Maintaining the requirement tracing during the project demands more time and effort, as does increased maintenance, particularly when the system is evolving, and significant changes are needed for next releases. There needs to be some managerial adjustment, especially with the relation

to developers. A propensity to be unsure of when to quit tracing.

[6] The capacity to meaningfully chronologically interrelate the uniquely identified items is known as traceability. Requirements traceability is a sub-discipline of requirements management in software development. It serves to ensure that the requirements defined for a system are duly tested and that the requirements are linked at every point during the verification process.

We have performed forward traceability on our prototype application.

User require ments	Functional requirements	Design element	Test cases
FRS-001	While supplying the ride posts, the app must search for rides based on the destination supplied and according to the rider's location.	Home catalog catalog.home()	TC-001 Searching ride posts based on the destination. TC002 Searching ride posts based on use
FRS-002	The App shall add up the travel time duration of all the users present locations, journey distances, and final destinations.	Home catalog catalog.Duration()	TC-003 Changing user present location and checking if there is any change in the travel time duration. TC-004 Changing the destination and checking whether the present travel time duration is changing or not.
FRS-005	The App must charge 50% of the total ride to the user who has cancelled the ride.	Ride catalog. catalog.cancellation()	TC-005 Cancelling the ride before the estimated pickup time. TC-006 Cancelling the ride at the end of the ride.
FRS-007	The app shall supply rewards based on a specific number of rides taken or rides provided, and the first-time user gets rewarded with a reward(provide rewards after user's 6th ride).	Reward catalog catalog.reward()	TC-007 Taking ride twice and posting ride four times and checking whether the rewards are provided or not. TC-008 Pretending to be a new user by registering with same email id.

Table 6 Requirement traceability

Conclusion

This paper focuses on Fudee application that has been developed to help commuters in the US. It studies how the requirements are gathered and enhanced. It then examines the forward traceability that has been performed on the application.

References

[1] https://www.way.com/blog/how-the-world-commutes/

[2]https://www.weforum.org/agenda/2022/05/commute-america-sustainability-cars/

[3] https://www.bts.gov/data-spotlight/record-breaking-increases-motor-fuel-prices-2022

[4] Lecture- Week 3 (03b_Req_Elicitation2.pptx slides)

[5] Lecture-Week-2b (functional_non_functional.pptx slides)

[6] Lecture-Week-10 (10a_Requirement Traceability.ppt slides)

[7] What Is Elicitation: Top Requirement Elicitation Techniques for 2023 (simplilearn.com)

[8] Requirements traceability - Wikipedia

[9] What is Requirements Traceability? - Requirements.com

[10] (PDF) Collaborative Requirements Elicitation: A Process-Centered Approach (researchgate.net)