**Motjoka Fanana ST10089515 SAND6221 Assignment 1**

**QUESTION 1**

The System Development Life Cycle (SDLC) could be a prepare utilized to plan, create, and keep up computer program frameworks. It's a system that guides designers through the complete prepare of building a framework, from arranging to sending and support.

The SDLC consists of a few center forms, each of which plays a vital part within the victory of the extend. These forms are:

Planning:

In this phase, the objectives, and destinations of the extend are set up, and a arrange is created to realize them. This incorporates characterizing the scope of the project, identifying partners, and deciding the assets required.

Significance:

Planning is basic to guarantee that the venture is well-defined, which all partners have a clear understanding of what is required. Ignoring this handle can result in destitute scope definition, which can lead to venture delays and taken a toll invades. For example, on the off chance that a company starts building an application without legitimate arranging, they may realize midway through the venture that the application doesn't meet wants of their target gathering of people.

Analysis:

This process includes gathering necessities from partners and analyzing them to decide the best approach for creating the framework.

Importance:

Analysis is basic since it makes a difference guarantee that the framework meets the wants of partners. Neglecting this handle can result within the improvement of a framework that doesn't meet the necessities of its clients. For case, in case a healthcare supplier executes a framework for understanding records without legitimately analyzing the prerequisites, the framework may not be user-friendly or effective, coming about in mistakes and delays in understanding care.

Design:

In this handle, the engineering of the framework is decided, and the specialized determinations are created.

Significance:

Plan is fundamental since it lays the establishment for the development of the system. Neglecting this prepare can result in a framework that's ineffectively planned, driving to technical issues and tall support costs. For case, if a company creates an online site without legitimately planning the client interface, clients may have trouble exploring the site, coming about in a destitute client encounter.

Implementation:

This prepare includes creating the framework agreeing to the specialized determinations.

Significance:

Usage is basic since it's where the framework is really built. Ignoring this handle can result in a framework that's buggy and untrustworthy. For illustration, in case a company surges the usage of a unused framework, they may introduce bugs that cause the framework to crash or information to be misplaced.

Testing:

In this prepare, the framework is tried to guarantee that it meets the requirements and capacities accurately.

Significance:

Testing is basic since it guarantees that the framework works as anticipated which any bugs are recognized and settled. Neglecting this process can result in a framework that's questionable and unreliable. For case, in case a company discharges an application without legitimately testing it, clients may experience bugs that cause information loss or security breaches.

Maintenance:

This handle includes keeping up the framework after it has been deployed, including fixing bugs and including unused highlights.

Significance:

Support is basic since it guarantees that the framework remains utilitarian and secure. Dismissing this handle can result in a framework that becomes out of date or defenseless to security dangers. For example, in case a company ignores to preserve their software package, it may end up obsolete and incongruent with more current innovation, making it helpless to security dangers.

In conclusion, the SDLC is a significant handle for guaranteeing that computer program frameworks are created and maintained properly. Each center handle plays a vital part in the victory of the extend, and ignoring any of them can have serious results, as outlined by the illustrations given over.

**QUESTION 2**

**Q.2.1)**

Senior community:

1. Older communities, who are the main beneficiaries of the application, should be considered as the main actors. Your input is critical to understanding their needs, challenges and preferences when arranging transportation for their errands. Conducting surveys, interviews, or focus groups with older adults can help gather valuable first-hand information and help shape application requirements.
2. Senior Associations/Associations:

Local seniors' clubs and organizations play an important role in representing and promoting the needs of older people. The participation of representatives of these groups gives the target user insight into the specific needs and expectations of the older community, based on her extensive experience and interaction with her group.

1. Caregiver/Family:

Caregivers and family members of older people often play an important role in managing their mobility needs. Their involvement as stakeholders provides valuable insight into the challenges and requirements faced in organizing transportation for seniors. These provide insight into ease of use, security concerns, and the need for coordination between applications and caregivers/family members.

1. Local transportation:

Working with local transport agencies, such as taxi companies, ride-sharing services, and municipal transport services, is key to understanding the potential for collaboration or integration with existing transport infrastructure. Your input influences the application's technical requirements, integration possibilities, and operational considerations to ensure compatibility and usability within the local transportation ecosystem.

1. Medical professionals:

Medical professionals such as doctors, nurses and pharmacists can provide valuable insight into the specific transportation needs associated with healthcare services. Their expertise helps identify requirements for doctor appointments, medication pickup, and coordination with medical facilities. Involvement of medical professionals ensures that the application adequately meets the specific needs of the medical practice.

1. IT Professional/Developer:

While you're the lead developer on the project, even if you're not directly involved in programming, having other her IT experts and developers on board can provide valuable input. Their views on technical feasibility, scalability, security considerations, and integration options help shape the requirements definition and ensure application viability and compatibility with existing IT infrastructure. increase.

**Q.2.2)**

1. User registration and profile management:

The application should have the ability for older users to create their own account and manage their profile. This will allow us to provide personal information and specific needs such as mobility requirements and communication preferences.

2. Transportation Service Inquiry and Planning:

Elderly users should be able to request errand transportation through the application. You should be able to specify the type of service you want. B. Go to the pharmacy or grocery store and schedule pick-up and drop-off times.

3. Driver matching and distribution:

The application needs a system that matches requested transportation services with available drivers. The system should consider factors such as driver proximity, availability, and specific needs of older users (e.g., need for wheelchair accessible vehicle). If a match is found, the system should dispatch the appropriate driver to perform the requested service.

4. Real-time tracking and notifications:

Older users should be able to see the progress and location of assigned drivers in real time. The application should provide up-to-date information about the driver's location and estimated time of arrival. Additionally, users should receive notifications such as text her messages and push notifications to keep them updated on the status of their transport requests.

5. Feedback and rating system:

The application should include a feedback and rating system that allows advanced users to share their experiences and provide feedback on transportation services. You should be able to rate drivers based on factors such as punctuality, professionalism, and overall quality of service. This feedback helps us improve our services and provides valuable insights for both drivers and development teams.

**Q.2.3)**

Use case: Plan your pharmacy visit.

Goal: Request a pick-up service for a pharmacy visit.

Description:

The user selects the "pharmacy visit" service, enters the desired pick-up/return date and time, and transmits the request. The system assigns an available driver to fulfill the request and confirms the scheduled transportation service.

Use case: plan shopping.

Goal: Request transportation for grocery shopping.

Description:

The user selects the "grocery shopping" service, enters the details of the desired grocery store, enters the desired pick-up/return date and time, and submits the request. The system matches requests with available drivers and schedules transportation services accordingly.

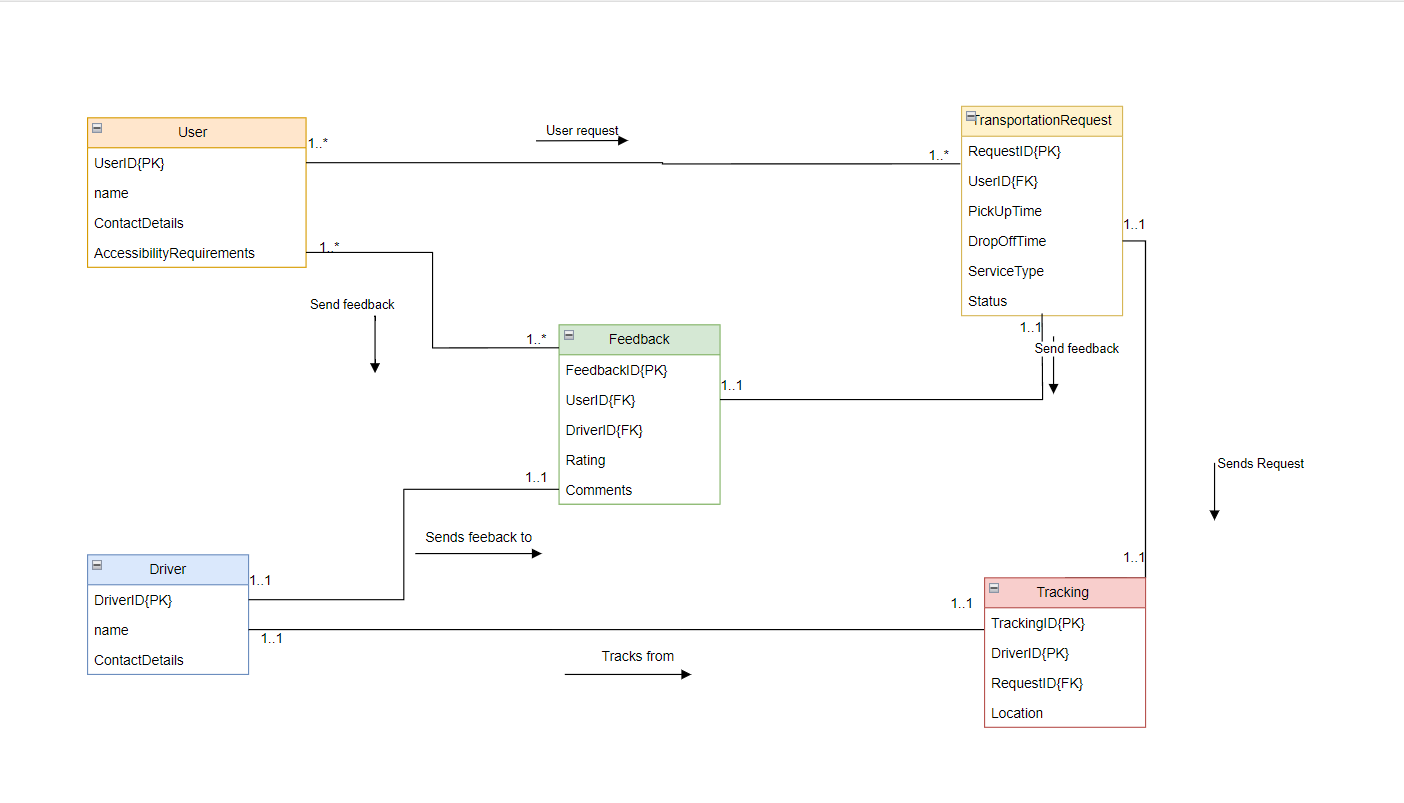
Use case: Schedule transportation to see a doctor.

 Goal: Request a pick-up service for a doctor's visit.

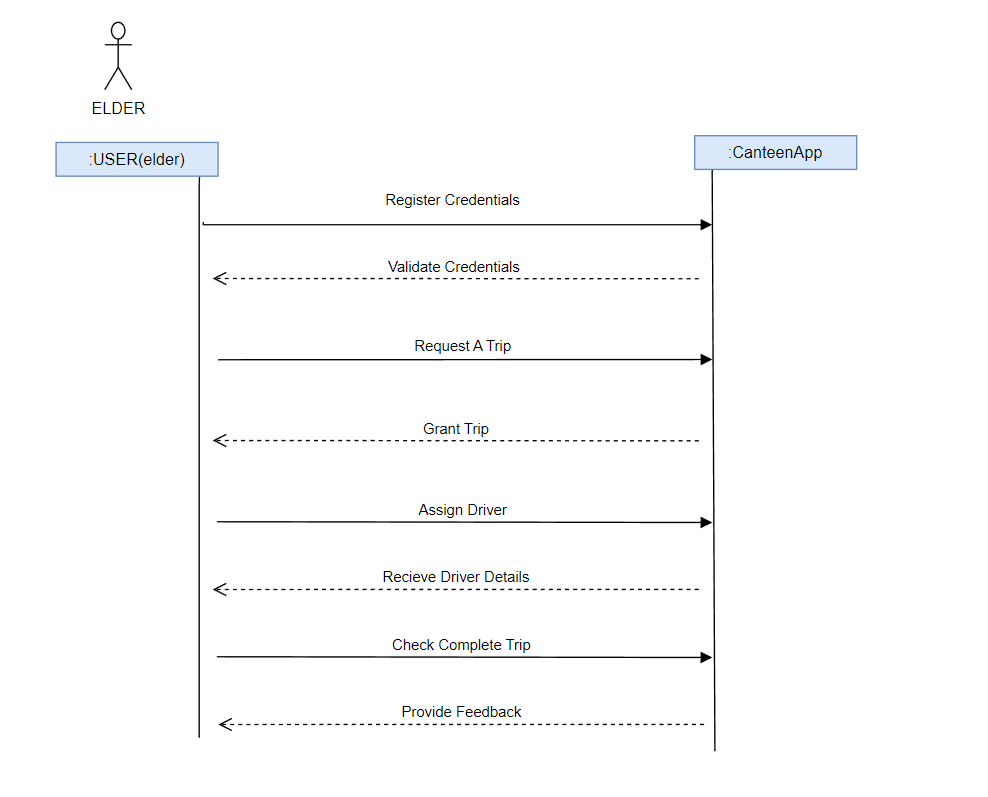
Description:

The user selects a medical appointment service, enters appointment details (clinic or hospital name, appointment time, etc.), and specifies a preferred pick-up time. Once you submit your request, the system will assign a driver that can meet your specific needs and schedule your transport service accordingly.

**Q.2.4)**



**Q.2.5)**



**QUESTION 3**

**User verification:**

There is a strong system in place to confirm a user's identity before granting them access to the application. This may include using strong passwords, requiring additional verification steps such as fingerprint or facial recognition, or using a combination of factors to ensure that only authorized persons can use the app.

**Data encrypt:**

Use advanced techniques to convert sensitive information into secret code, both when transferring that information between the application and the user and when storing it in the database. This way, even if someone intercepts the data, they won't be able to understand the data without the encryption key.

**Secure input management:**

Develop thorough verification and cleaning mechanisms of all user-supplied information to prevent potential attacks. By carefully reviewing and cleaning user input, you can avoid common security risks such as malicious code insertion or unauthorized access to parts of your application.

**Updated frequently:**

Stay alert to any updates or security patches released by the software and tools you use for your apps. Make sure to install these updates in time to fix any vulnerabilities that may be discovered. Regularly updating apps keeps apps safe and protects them from new threats.

**Incident monitoring and response:**

There is a system in place to monitor any unusual activity in the application. By monitoring and recording important events, you can quickly identify suspicious behavior or security breaches. This allows you to act immediately and investigate any potential problems before it causes major damage.

**Question 4**

**Q.4.1)**

**Cooperation and communication:**

Agile development emphasizes close collaboration and effective communication between team members, stakeholders, and end users. Foster a collaborative environment where everyone is encouraged to actively participate and contribute throughout the development process. Regular meetings, such as daily face-to-face or sprint improvements, can facilitate open communication and promote transparency within the team. By embracing a collaborative culture, you can ensure that everyone's ideas are considered, leading to better results and a stronger sense of cohesion among team members.

**An iterative and incremental development:**

Agile methods, such as Scrum, favor iterative and incremental development. Instead of trying to complete an entire project at once, break it down into smaller, manageable chunks called sprints. Each sprint focuses on providing a functional and useful part of the application. Adopting this approach allows for regular feedback and validation from stakeholders and end users. It allows you to quickly adapt to changing needs, refine your product, and make improvements based on user feedback. By iterating and gradually delivering functionality, you can ensure that the project evolves according to the needs and expectations of the users.

**Flexibility and adaptability:**

Agile development accepts change and recognizes that requirements and priorities can change throughout the project. Embrace the flexibility to meet changing needs and adjust your plan accordingly. Nurture a customer satisfaction mindset through early and ongoing delivery of valuable software. This means being open to feedback, adjusting, and re-prioritizing tasks as needed. By embracing flexibility and adaptability, you can ensure that your project stays relevant to the changing needs of your target audience.

**Continuous improvement and reflection:**

Agile methods foster a culture of continuous improvement. Regularly reflect on team performance, processes, and results. Make improvements after each sprint to identify strengths, weaknesses, and areas for improvement. Encourage team members to openly share their ideas and suggestions to improve collaboration, productivity, and product quality. Act on these results by making changes and testing new methods. By fostering a culture of continuous improvement, you create an environment where the lessons learned are appreciated and where the team can grow and mature together.

Q.4.2)



**REFERENCE**

Gillis, A.S. (2019) *What is systems development life cycle? - definition from whatis.com*, *Software Quality*. Available at: https://www.techtarget.com/searchsoftwarequality/definition/systems-development-life-cycle (Accessed: 17 May 2023).

Harsha, S. (2019) *Usability testing in design - why is it important?* *Medium*. Available at: https://uxdesign.cc/usability-testing-in-design-and-why-is-it-important-cfddfbbdaac9 (Accessed: 19 May 2023).

*What is user centered design?* (2023) *The Interaction Design Foundation*. Available at: https://www.interaction-design.org/literature/topics/user-centered-design (Accessed: 8May 2023).

Fanana, M. (2023) ‘Diagram.io’, *diagram.io*. Available at: diagram.io (Accessed: 8 May 2023).

*UML use case diagram tutorial* (no date) *Lucidchart*. Available at: https://www.lucidchart.com/pages/uml-use-case-diagram (Accessed: 11 May 2023).

*UML models and diagrams* (2021) *IBM*. Available at: https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-uml-models (Accessed: 19 May 2023).

*Entity relationship diagram* (no date) *Entity Relationship Diagram - Data Modeling - UML Diagramming Software*. Available at: https://www.visual-paradigm.com/VPGallery/datamodeling/EntityRelationshipDiagram.html (Accessed: May 17, 2023).

*UML - Basic notations* (no date) *Tutorials Point*. Available at: https://www.tutorialspoint.com/uml/uml\_basic\_notations.htm (Accessed: May 17, 2023).

Coronel, C. *et al.* (2020) *Database principles: Fundamentals of design, implementation, and management*. Australia: Cengage.