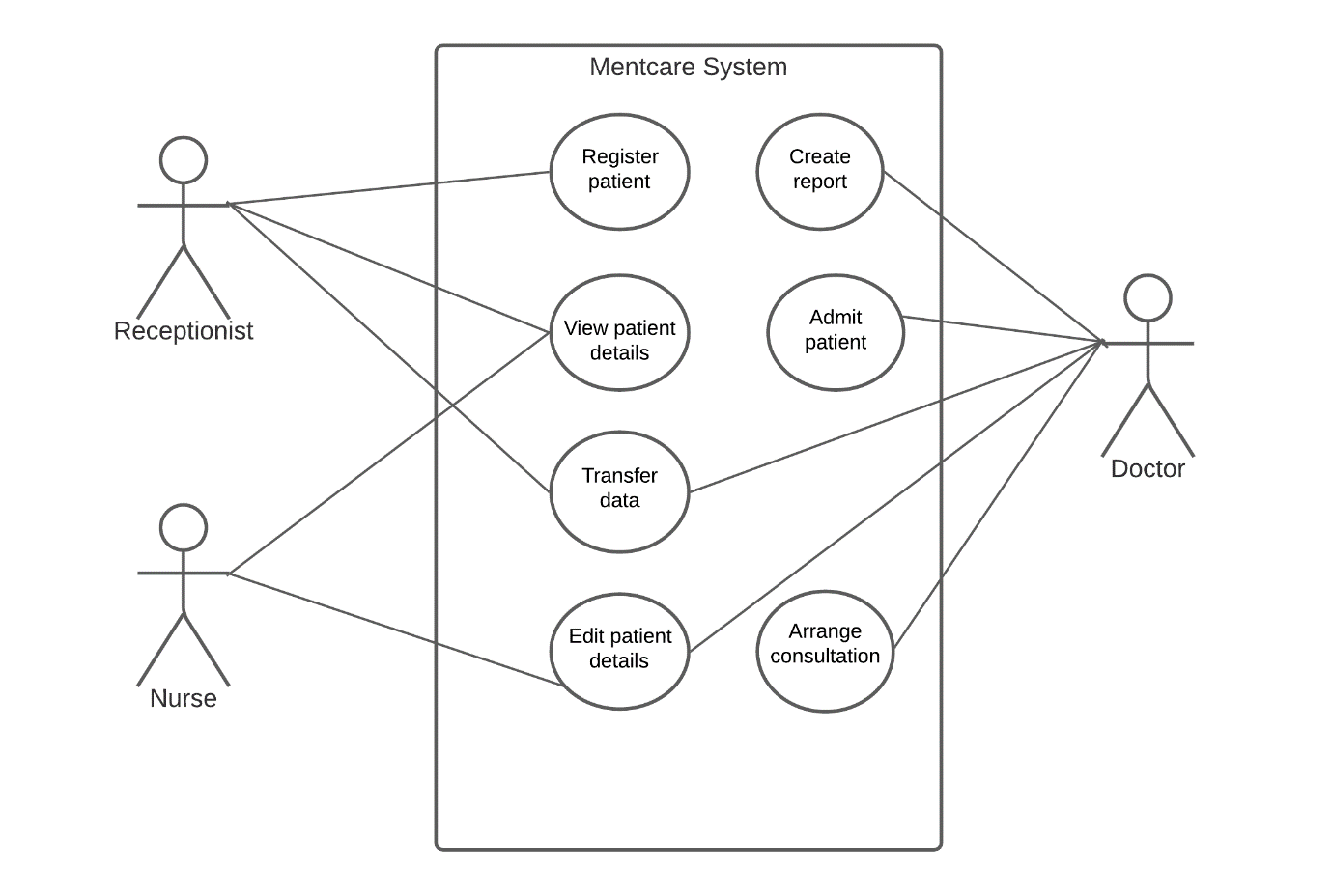
DCIT 208 FINAL ASSIGNMENT

**10841867**

1. A typical information system is made up of entities of which have externally visible properties. Software architecture deals with the structure of the structures of this information system and the relationship that exists among them. The discipline in organizing and establishing these vital structures and systems involves software architecture.

2. Use case of a men care system

3. **For**: Developers are responsible for their codes’ quality. When developers test their codes, they can confirm that the code, even after making changes still functions properly. Sooner or later, doing this helps save time as issues and downtime on production are greatly reduced.

**Against**: Some teams require developers to create automated and integrated tests that are code-based. But the development of these tests consumes time, time of which could be spent writing new codes. Also, separate teams responsible for testing codes are so familiar or used to subtle variations of application systems making them very efficient or competent at testing. These teams improve the success of tests of codes as they are impartial and uninfluenced and provide the human element needed. The conclusion is that developers running code-based tests slows down the development process as well as the productivity of codes.

4. As time flows, new requirements appear and the existing ones also change. This is due to the changes in the business running the software. Errors need to be dealt with, the addition of new features are made and non-functional requirements such as performance must improve as other competitors do the same. Therefore, software systems should evolve to adapt to varying environments. When software systems remain the same even as the environment changes, users migrate to other software systems that can meet their emerging requirements, making the software systems less helpful or useful.

5. A program can be correct and also exhibit bad quality. When a program receives input from a user and it displays or produces the right and expected result, it is correct. But when the program crashes when given an input that wasn’t taken into consideration during the development of such a program, it is not reliable and does not exhibit good quality. It, therefore, produces unwanted output. If the program also takes way too long to produce a result, it is correct (as long as the result is right and expected) but it also exhibits bad quality.

6. Systems designed to have a high level of availability are accompanied by high requirements. A lot of people are going to use this system and therefore it should be available anytime. It should also meet the requirements. Since a lot of people will be involved in such a system, occurrences of errors as well as the system operating wrongly is highly not tolerated. Therefore, all exceptions in this system should be explicitly handled to achieve high availability. Systems must also recover automatically when an error happens.

7. Assessment of software quality is possible even with its requirements changing. Changing requirements are common in Agile practice and assessment of software quality can be done. Assess it by looking at both the current features and functionality of the software and its (changing) requirements. The more the features and functionality match the requirements, the better the software’s quality.