GERMAN INTERNATIONAL UNIVERSITY
الجامعة الألمانية الدولية

FACULTY OF INFORMATICS
Dr. John FW ZAKI
Sunday, 03.11.2023

REQUIREMENT ENGINEERING
Winter Semester 2024/2025
Midterm Exam

Bar Code

Instructions: **Read Carefully Before Proceeding**.

1. No **programmable** calculators, book or other aids are permitted for this test.
2. Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem.
3. Attempt as many of the problems as you can within the time limits. The more you solve the higher your score is expected to be.
4. Read all the problems carefully before starting and start with the easiest question you find.
5. This exam booklet contains **10 pages**, including this one. Extra sheets of scratch paper is attached and have to be kept attached.
6. When you are told that time is up, stop working on the test.
7. Total time allowed for this exam is **2 hour**.

Good Luck!

Please, do not write anything on this page

Question Number	**1**	**2**	**3 BONUS**	**Total**
Obtained Score				
Maximum Score	**60**	**40**	**5**	**100**

Q1. [60M - 60 MIN]

Mark:	/60

In 2008, Netflix (an online streaming company) started transitioning from a traditional data center to AWS cloud. The story is one of the most notable cloud migration stories shedding light on scalability, flexibility, and fault tolerance. As their customer base started to grow, the operation became increasingly challenging.

Netflix started their journey in the 90's with a regular client server architecture with a single Oracle DB on a very large processor and a monolithic application. They used to pay huge Oracle DB fees. They had a few servers. Every year they used to increase their disc storage and compute power. Thus, they used to pay like 8x as much money they paid in previous year to just get 2x as much performance as last year. They wanted to ensure if they pay 8x as much money, they get 8x as much performance. Their app was a monolithic Java app talking to Oracle DB. They had around 200 engineers working on the system. They had a strategy of growing to almost 2000 engineers in 10 years' time. It was going to be impossible to manage the integration of 2000 engineers work into the monolithic app. This wasn't going to fly with the future vision. They wanted the engineers to be able to contribute to decoupled components. Thus, they build a cloud native application based on microservices approach and failover strategies to ensure resilience. Additionally, the system had issues with security, connection pools, and threads. Their SQL reached the point that it was extremely difficult to manage. Although SQL is very comfortable to do, it is very expensive to scale. They once updated their firmware which resulted in database corruption that took around 3 days to recover. They wanted something more flexible so when the application scales, they don't have to change schema or spend too much money on computational power. Hence, they resorted to NoSQL DB to reduce the issues with consistency, maintenance…etc. In terms of the application, they wanted to build something that had no architectural limits in order to move forward. They wanted to ensure that they had redundancy, flexibility, availability, and performance to enable their application to scale into the future. They had three different approaches to move their operations to the cloud.

1. Take the existing app and refine it, then put it on the cloud (unappealing)
2. Build from scratch and gradually move traffic to the new system (risky)
3. Migrate features one at a time where the old and the new system are running in parallel (The solution)

Eventually, they were able to move almost every feature to the cloud by 2014 except the payment and billing as they had issues with auditors and compliance. Eventually, they unplugged their physical datacenter in 2015.

Note: AWS is Amazon cloud services. They provide cloud infrastructure and help companies by taking some of the infrastructure heavy lifting, security, and maintenance out of the hands of the business.

1. **[6M] Identify 4 potential problems that Netflix faced with their initial architecture and situation. What could be the result of each of those problems?**

    a. **Monolithic App → Inability to scale, Single point of Failure**
    b. **Single DB → single point of failure**
    c. **Difficulty of integrating work of engineers**
    d. **Issues with consistency, maintenance, flexibility, availability….etc.**
    e. **High operational cost**
    f. **System vulnerabilities**

    Each name 1 mark, the result of each is half mark

2. **[4M] What can Netflix do to improve their agility in the software?**

    a. **Breakdown the monolithic App → Design a cloud native application**
    b. **Use agile methodology in their application development**
    c. **Implement CI/CD in their application development**

    Any one answer can be correct and the focus on the first one.

3. **[5M] What is Netflix AS-IS model for the situation (write in points or plot)?**

- **Monolithic Application**
- **Single Oracle Database**: leading to single point of failure
- **Client-Server Architecture**
- **High Cost and Low Scalability**
    - **Difficult integration points**

4. **[5M] Suggest a TO-BE model that fits with Netflix Strategy (write in points or plot)**

• **Microservices Architecture**: Decouples the monolithic application into independent, small services, each handling specific functions (e.g., user management, recommendations).

- **Failover Strategies**

• **NoSQL Databases**

• **AWS Infrastructure**

- **High Scalability and Availability**

Agile approach to ensure flexibility.

5. **[6M] List one non-functional external requirement, 4 non-functional product requirements, and one non-functional organizational requirement.**

Non-Functional External Requirement:

1. **Data Regulations:** The system must ensure compliance with external regulations, and data privacy laws to protect user data.

Non-Functional Product Requirements:

1. **Scalability**
2. **Availability**
3. **Performance**
4. **Security**

Non-Functional Organizational Requirement:

1. **Team Collaboration and Efficiency**

6. **[8M] Netflix faced many issues with their initial model, you are the RE engineer tasked with analyzing the root causes of the issues. Use fishbone (Ishikawa) Diagram to categorize and organize those challenges (use minimum 4 categories).**

1. People

- **Skill Gap**
- **Coordination Issues (integrating 2000 engineers work together)**

2. Process

- **Lack of Agility**
- **Manually doing certain tasks.**

3. Technology

- **Monolithic Architecture**
- **Database Scalability Issues**

4. Environment

- **Rapid Growth in User Base**
- **High Cost**

Any other ideas around the same lines can be accepted

7. **[4M] From your RE experience, discuss the requirements for Netflix scalability.**

Write about the DB and how the NoSQL is important and write about the APP and how they should break it down to microservices architecture and use the cloud native application.

8. **[4M] What sort of elicitation technique would you use to ensure smooth operation of "Migrate one feature at a time"? Why did you choose this approach?**

System Interface Analysis

Because the two systems (old one and new one) must integrate with each other to enable seamless user experience.

9. **[4M] What were the advantages and risks associated with Netflix incremental migration approach?**

Advantages:

1. **Reduced Risk of System-Wide Failure**
2. **Better Control and Flexibility**
3. **Minimized Downtime**
4. **Easier Testing and Validation**
5. **Enable parallel operation and development**

Risks:

1. **Increased Complexity**
2. **Longer Migration Time**
3. **Resource Intensive**
4. **Integration Issues**
5. **Data consistency**

Anything around those lines can be accepted.

10. **[4M] Discuss how requirement prioritization played a role in migrating features gradually. Mention a simple and fast prioritization technique that can be used prioritize features.**

    (IN OR OUT)
    OR
    (MoSCoW)

11. **[10M] Given the three migration approaches above considered by Netflix, use value-risk & cost model (Wieger's prioritization Matrix) to <u>prioritize the 3rd approach (gradual feature implementation)</u>. Once you have finished the calculations insert the approaches into the table to reach the same recommended solution as Netflix. Ensure you show your calculations under the table.**

Relative Weight	**2**	**1**			**1**		**1**		
Gradual feature implementation	**2**	**4**	**8**	**16.67**	**1**	**11.11**	**1**	**16.67**	**0.6**
Build from scratch	**9**	**7**	**25**	**52.08**	**5**	**55.55**	**3**	**50**	**0.49**
Refine the current version	**5**	**5**	**15**	**31.25**	**3**	**33.33**	**2**	**33.33**	**0.46**
			48		**9**		**6**		

Q2. [40M - 40 MIN]

Mark:	/40

1. **[20M] You are designing a smart parking system for a city. The system tracks available parking spots in real-time and directs drivers to the nearest available spot through a mobile app. The parking spots are equipped with sensors to detect whether they are occupied or not, and the system also integrates payment processing for parking fees.**

Create a simple WRSPM reference model for this smart parking system, defining the hidden and visible elements of the environment and system.

W: (World Assumptions)
R: (Requirements)
S: (Specifications)
P: (Program)
M: (Machine)

World:

1.Parking Spots

Requirements:

1. **Real-time Spot Detection**
2. **Driver Guidance**
3. **Payment**
4. **User Notifications**

Specifications:

1. **Sensor Data Collection**
2. **Driver Guidance Algorithm**
3. **Payment Integration**
4. **Notification Service**

Program:

1. **Mobile App**
2. **Backend Server**
3. **Payment Gateway**

Machine:

1. **Parking Spot Sensors**
2. **Network Infrastructure**

EV: Vehicles EH: Traffic Conditions SV: Mobile Application Interface SH: Driver Guidance Algorithm

2. **[20M] Your team is tasked with developing a personalized fitness tracking and workout recommendation app for users of all fitness levels. The app should allow users to log their workouts, track their fitness progress over time, and receive notifications for new workout routines based on their goals. Additionally, the app will offer personalized workout suggestions based on user preferences and activity history. Users should also be able to rate and review workout plans to help others in the community.**

Identify and describe at least 7 user requirements. (neglect the acceptance criteria in correction)
Write them in the form of a user story. "As a user, ……. "
Extend your answer to indicate the acceptance criteria for each requirement.

1. **As a user, I want to log my workouts easily so that I can track my exercise routine and monitor my fitness progress over time.**
    a. **Acceptance Criteria: The app should save the logged workout data and display it on the user's profile.**

2. **As a user, I want to view my fitness progress over time so that I can see my improvements and stay motivated.**
    a. **Acceptance Criteria: The app should provide a comparison of current and previous workout performance.**

3. **As a user, I want to set fitness goals and receive notifications when new workout routines align with my goals so that I stay on track.**
    a. **Acceptance Criteria: The user can set specific fitness goals**

4. **As a user, I want to receive personalized workout recommendations based on my fitness level and past workouts so that my routines align with my personal goals and interests.**
    a. **Acceptance Criteria: The app should analyze the user's workout history and preferences to suggest suitable workout routines.**

5. **As a user, I want to rate and review workout plans so that I can help other users in the community find good routines.**
    a. **Acceptance Criteria: The user can rate a workout plan on a scale (e.g., 1-5 stars).**

6. **As a user, I want to view workout reviews and ratings from other users so that I can select highly recommended routines.**
    a. **Acceptance Criteria: The user can filter workout plans by rating and popularity.**

7. **As a user, I want to sync the app with my wearable fitness devices (like a smartwatch) so that my workout data is automatically imported.**
    a. **Acceptance Criteria: The app should automatically update and display real-time data for metrics such as heart rate, steps, and calories burned.**

Q3. [5M - 20 MIN] BONUS QUESTION

Mark:	/5

SmartCampus will provide an integrated platform for students to view their schedules, receive campus announcements, access digital resources, and connect with campus facilities. With limited resources, only a subset of features can be completed for the initial release.

The following features have been proposed for the initial release.

1. **Personalized Student Timetable** – Display each student's class schedule, including class location and time changes in real-time.
2. **Campus News Feed** – Provide a feed of important campus news and events to keep students and academics updated.
3. **Digital Library Access** – Allow students to search and download resources from the campus library.
4. **Emergency Alerts** – Send push notifications for emergency updates (e.g., severe weather, campus security alerts).
5. **Social Clubs Integration** – List student clubs, allow users to join, and receive club updates.
6. **Assignment Tracker** – Track upcoming assignment deadlines and notify students of changes.
7. **Campus Map with Navigation** – Show interactive campus maps and directions to navigate between buildings.
8. **Faculty Directory** – Provide contact information for faculty members and office locations.
9. **Study Room Booking** – Enable students to book study rooms in the library.
10. **Cafeteria Menu and Pre-Ordering** – Show daily cafeteria menus and allow students to pre-order meals for faster pickup.
11. **Campus Bus Schedule** – Display real-time bus schedules for campus round buses.
12. **Student Feedback System** – Allow students to submit feedback on classes, campus facilities, and events

Prioritize the features that will be developed in each sprint of the release using MoSCoW. Provide a justification for the prioritization.

Personalized Student Timetable	M	Essential for students to keep track of their classes.
Campus News Feed	M	Keeps students informed about campus events and critical updates.
Emergency Alerts	M	Crucial for student safety and compliance with campus security needs.
Digital Library Access	S	Important for academic resources but not essential for initial release.
Campus Map with Navigation	S	Useful for students, especially new ones, but not critical.
Faculty Directory	C	Convenient but not critical to primary functionality of the app.
Study Room Booking	C	Beneficial but not as essential as other features in initial rollout.
Assignment Tracker	S	Helps students manage deadlines but not a core function.
Social Clubs Integration	W	Can be deferred, not essential for initial campus navigation.
Cafeteria Menu and Pre-Ordering	W	Adds convenience but can be left for later development.
Campus Bus Schedule	C	Nice to have but not core functionality.
Student Feedback System	W	Useful for improvements but not crucial for app's initial purpose.