# NegevNerds

## Collaborative Exam Preparation Platform

## Application Requirements Document


Ben-Gurion University of the Negev

Software Engineering Department

**Project Advisor:**

Dr. Elior Sulem

**Team Members:**

Shachar Cohen

Nadav Ktalav

David Volodarsky

Noa Aboody


## Abstract

NegevNerds is an online collaborative website designed to assist students at Ben-Gurion University in preparing for exams. The platform allows students to ask questions, discuss past exams, creating a valuable repository of exam content. NegevNerds aims to foster peer-to-peer collaboration and provide an easily accessible, interactive space for exam preparation.

# Contents

# 1. Introduction

## 1.1. The Problem Domain

Students today face significant challenges in efficiently preparing for exams. Although previous exams are accessible through platforms like “Google Drive”, the lack of an effective search function makes it difficult to find specific questions by date or keywords. As a result, students waste valuable time searching for the questions they want to practice.

In addition, students often rely on “WhatsApp” groups for discussing answers to exam questions. However, the management of information within these groups is disorganized, making it hard to access relevant data or follow past discussions. This leads to inefficiency and poor knowledge retention.

The domain of this project centers on **collaborative exam preparation**, with the goal of creating a platform that supports the collection, organization, and sharing of past exam questions and their answers.

The project’s main problem domain involves:

- **Efficient search functionality** finding exam questions by date, topic, or keywords, allowing students to quickly find the material they need.
- **Knowledge retention** through organized discussions of answers to questions, making it easy for students to revisit and engage with prior conversations.
- **Collaborative engagement and communication** with other students, enabling students to support one another through shared content and peer-to-peer communication.

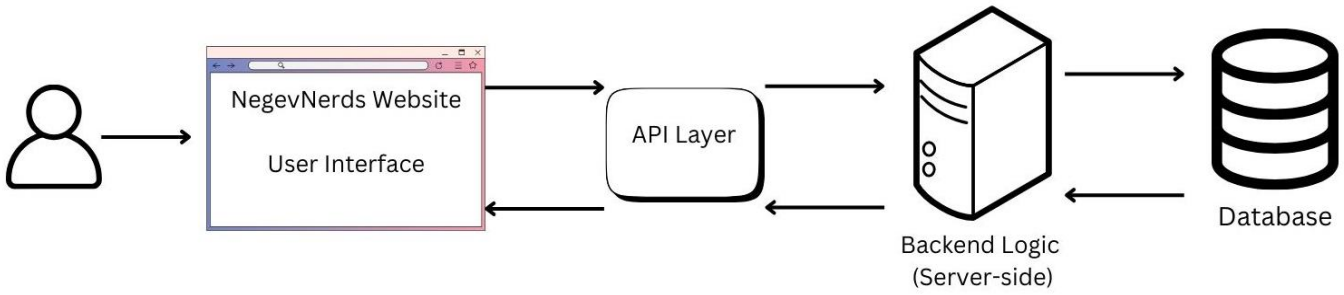
## 1.2. Context

The NegevNerds system operates within the context of an online platform designed to support collaborative exam preparation for students at Ben-Gurion University. The system enables students to engage in discussions around exam questions, share resources, and create a searchable database of questions and answers. This collaborative environment is intended to foster peer-to-peer learning and provide a repository of useful materials for exam preparation.

The system interacts with the university's existing infrastructure, such as student authentication systems, and integrates with an online database to store user-generated content like questions, answers, and discussions.

The major components of the system and their interactions are described below.

Block Diagram of the NegevNerds System:




## 1.3. Vision

The main goal of our project is to develop a platform that revolutionizes the way students prepare for exams by providing a collaborative and organized environment for sharing, searching, and discussing past exam questions. We aim to eliminate the inefficiencies caused by fragmented resources, such as unorganized WhatsApp groups and poorly structured shared files. Our platform will allow students to easily search for questions by specific topics, keywords, or dates, making exam preparation more focused and efficient.

Through this system, students will not only have quick access to relevant questions and answers but will also be able to engage in organized discussions, contributing to the collective knowledge of the student community. By fostering collaboration, the platform will help students retain knowledge more effectively and streamline their study process.

In the future, as we address the challenges of organizing and sharing exam materials, we envision expanding the platform's capabilities to include more interactive features, such as private chats and even mobile app integration, ensuring a complete and comprehensive study experience for all students.

## 1.4. Stakeholders

**Students at Ben-Gurion University:** The primary customers and users who will use the platform for exam preparation, searching for past exams, and participating in discussions. Students will also play a crucial role in contributing to and maintaining the platform by uploading new exams, sharing study materials, and engaging in collaborative discussions to enrich the content and ensure the platform remains current and valuable for future users.

## 1.5. Software Context

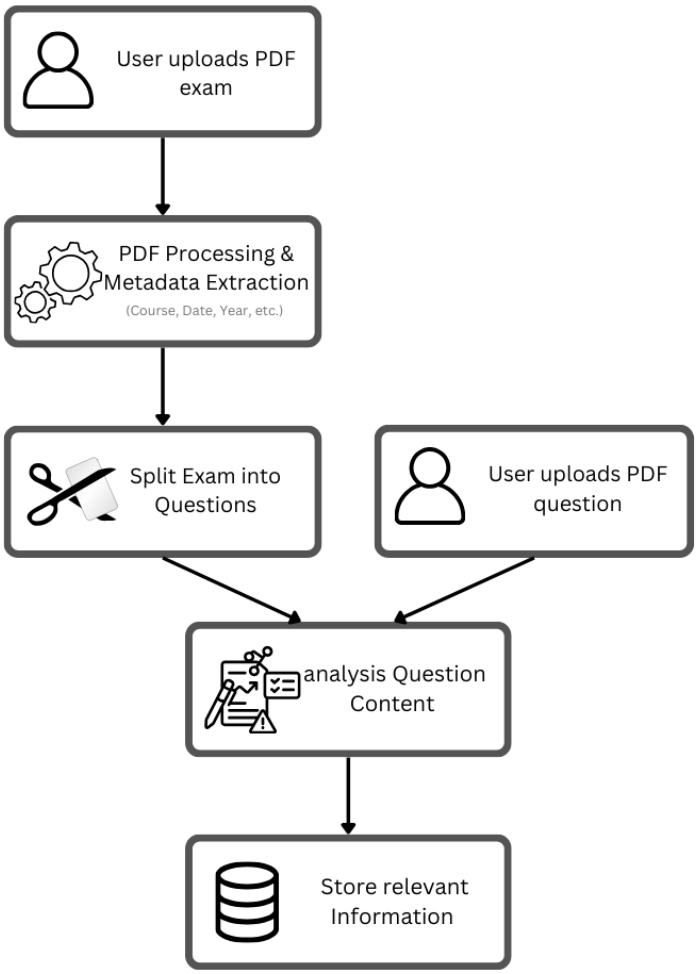
### 1.5.1. Exam Management Unit

This unit is responsible for managing and organizing the past exam questions uploaded by users. It allows the system to categorize and store content in a way that facilitates easy searching and retrieval. The unit processes incoming exam questions by extracting relevant metadata (e.g., course name, date, topic) and associates them with the appropriate tags or categories.

#### 1.5.1.1. PDF Management Unit:

The user uploads an exam PDF, and the system analyzes the PDF to extract the necessary information, such as the course name, year, exam session, and splits the exam into individual questions.

1.5.1.2. Question Analysis and Indexing:

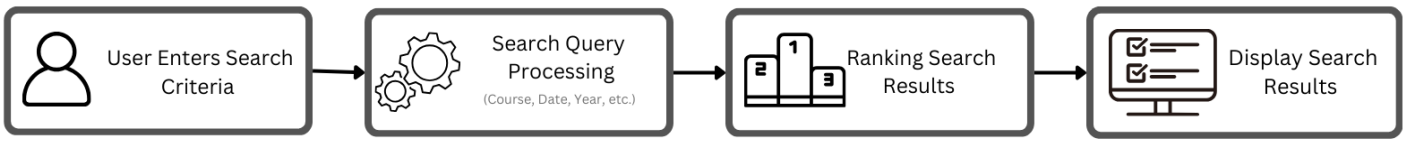



Once the questions are split from the PDF, they require further analysis so the system can perform accurate search. This is where **Information Retrieval and Indexing** comes into play.

1.5.2. **Search and Filtering Unit**

The Search and Filtering Unit is responsible for allowing users to **search for exam questions** based on specific criteria, such as date, topic, or keywords. This unit processes the search queries and quickly returns the most relevant results. By utilizing advanced indexing and filtering methods, it ensures that students can easily find the exam questions and answers they need. The

system uses algorithms to rank and present search results in a user-friendly manner. Once the user enters search criteria, the unit processes the query



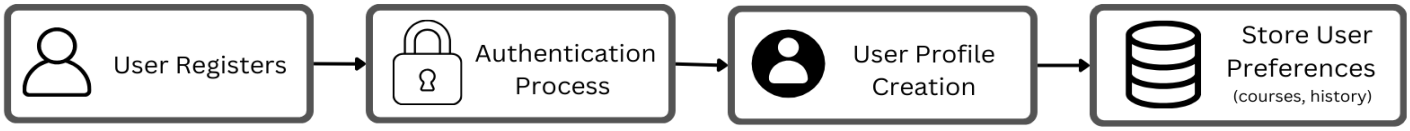

and matches it with the available database, providing accurate and timely results.

1.5.3. **Discussion and Collaboration Unit**

This unit supports peer-to-peer communication by enabling students to participate in discussions related to specific exam questions. The unit is responsible for creating and managing discussion, where students can **comment**, share insights, and ask questions. It also allows students to **upload files** (e.g., images, documents) to share relevant study materials or additional context related to specific exam questions and support **comment rating**, enabling students to rate answers or comments based on their usefulness or accuracy. The Discussion and Collaboration Unit provides an interactive space where students can engage with their peers, improving their understanding of exam topics.

1.5.4. **User Authentication and Profile Management Unit**

This unit handles user registration, authentication, and profile management. It ensures that students can securely log into the system using their university email addresses, and it maintains their personalized study history and preferences. The unit is responsible for creating and updating user profiles, which store information such as previously searched questions and participation in discussions.



### 1.5.5. Notifications and Alerts Unit

The Notifications and Alerts Unit is responsible for informing users about new content, discussions, and other relevant updates. It can send notifications when new exams are uploaded, when there are responses to a discussion they are following, or when new features are available. The system can notify users through in-app notifications and email, ensuring that students stay up to date with the platform's latest activities. This unit is crucial for keeping users engaged and informed about new information.

# 2. Usage Scenarios

## 2.1. User Profile – The Actor

### 2.1.1 Guest

A Guest refers to a user who has not registered or logged into the system yet. The guest interacts with the system in a limited way. The guest is not able to watch any contents on the website for security and copyrights reasons.

- Capabilities:
  - Register as a new user.
  - Log in with an existing account.
- Interactions with the System:
  - Sends registration or login data to the system.

### 2.1.2 User

A User is an individual who has created an account in the system. This actor interacts with the platform to access its core functionalities, including searching for and contributing to the exam preparation database.

- Capabilities:
  - Access and interact with all general site functionalities.
  - Search for exam questions using keywords, topics, or dates.
  - Register for courses within the system.
  - Open new courses.
  - Post questions or entire exams to the platform.
  - Comment on discussions related to posted questions.
  - Rate comments for their usefulness or relevance.
  - Participate in collaborative discussions with peers.


- Report on comments / questions.

- Interactions with the System:
  - Sends data related to course registration, questions, comments, and ratings to the system.
  - Receives search results, question content, discussion updates, and other collaborative contributions

### 2.1.3 Course Manager

A Course Manager is a user with additional responsibilities and privileges for managing course-specific content. This actor ensures the quality and relevance of contributions within their assigned courses.

- Capabilities:
  - All functionalities available to a User.
  - Remove or edit questions and exams within their course.
  - Moderate discussions to ensure constructive and relevant conversations.
  - Appoint new course manager.
  - Handle user's reports.
- Interactions with the System:
  - Sends data to modify or delete course-specific content (e.g., questions or exams).
  - Receives feedback or confirmation from the system regarding these changes.

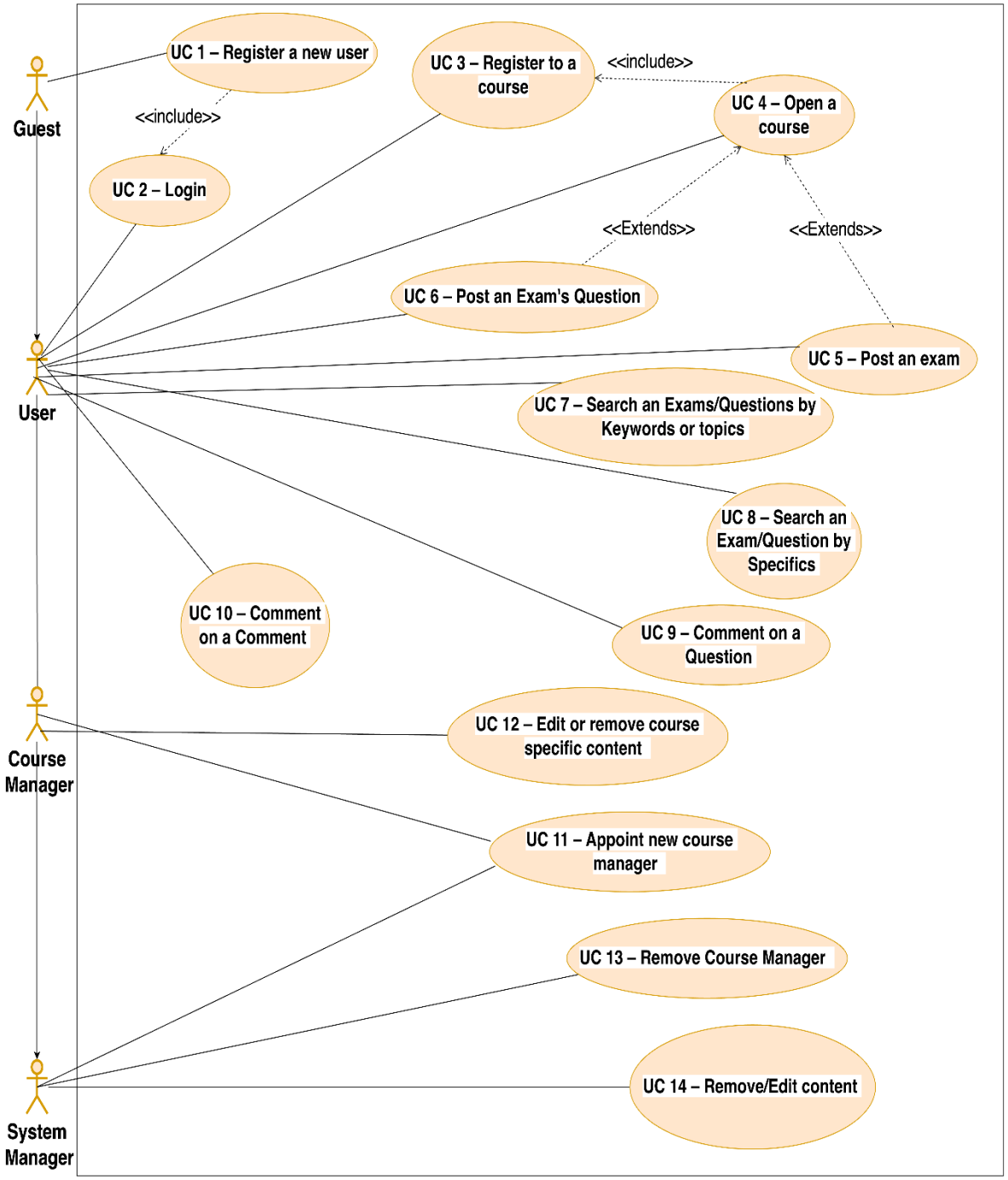
### 2.1.4 System Manager

A System Manager is the highest-level user, responsible for the overall administration of the platform. This actor has access to all functionalities of the system and additional administrative privileges to manage users and courses.

- Capabilities:
    - All functionalities available to a Course Manager.
    - Appoint or remove Course Managers for specific courses.
    - Oversee and manage the platform’s overall functionality, including system-wide moderation.
- Interactions with the System:
    - Sends data to appoint or revoke Course Manager privileges.
    - Monitors system logs and user activities for quality assurance.
    - Receives notifications and system status updates.

## 2.2. Use-cases

Use Cases Diagram:



### 2.2.1 UC 1 – Register a new user

Description: A guest is trying to access the platform. This is the guest first time in the platform therefore he will have to register before getting an access to the website.

Actor: Guest

Parameters: name, last name, mail address, password

Pre-conditions:

1. The guest has no access to the website content since he is not registered yet.

Post-Conditions:

1. The guest details are being save in the system database.
2. The user receives a confirmation email with registration details.
3. The guest, which is now a user, is being transported to the homepage of the website.

Main success scenario:

1. The guest enters the “NegevNerds” website.
2. The guest will fill the following details – name, last name, mail (BGU mail only) and password.
3. The guest will have to accept the platform terms of use.
4. The system validates the information provided.
5. A verification code will be sent to the guest email to verify is identity.
6. The guest will type the verification code that he got in his mail.
7. The system checks the provided code:
8. For a matching code, the registration will end successfully with the guest details are stored in the system database and the guest has become a user.

Alternative flows:

1. The guest is trying to register with a mail address that already in the system database - the system returns a relevant error message and registration fails.
2. The guest is trying to register with a non-BGU mail address, the system returns a relevant error message and registration fails.
3. The guest typed an unmatching verification code, the system returns a relevant error message and registration fails.

### 2.2.2 UC 2 – Login

Description: A user wants to log in to the platform to access its features and content.

Actor: User / Course Manager / System Manager

Parameters: email, password

Pre-conditions:

1. The use is not logged in.

Post-conditions:

1. The user is authenticated and redirected to their homepage.
2. User's login session is active.

Main success scenario:

1. The user navigates to the "NegevNerds" login page.
2. The user enters their email and password.
3. The system validates the credentials:
    a. If valid, the system grants access and redirects the user to their homepage.
    b. The user's session begins.

Alternative flows:

1. **Invalid Credentials:** If the entered email or password is incorrect, the system displays an error message: *"Invalid email or password. Please try again."*
2. **Forgot Password:** The user clicks "Forgot Password?" and is redirected to the password recovery flow.

### 2.2.3 UC 3 – Register to a Course

Description: A user wants to register for a course on the platform to access its content and participate in discussions.

Actor: User / Course Manager / System Manager

Parameters: course name / course ID

Pre-conditions:

1. The user is registered and logged into the platform.
2. The course exists in the system.

Post-Conditions:

1. The user's details are added to the course's participant list.
2. The course appears now in the user homepage.

Main success scenario:

1. The user search for a specific course by its name or id.
2. The user clicks on the register bottom of the course.
3. The system confirms the registration by displaying a success message.
4. The course can be seen on the user homepage now.

Alternative flows:

1. **The user is already registered for the course:** The system displays an error message stating the user is already enrolled.
2. The course is not available on the site: The system displays an error message stating that there is no course such course in the platform and suggest the user the option to open it.

### 2.2.4 UC 4 – Open a Course

Description: A user wants to create and open a new course on the platform.

Actor: User / Course Manager / System Manager

Parameters: course id, course name, course syllabus file

Pre-conditions:

1. The user is logged into the system.

Post-Conditions:

1. The new course is added to the system and is visible to students.
2. The user who opened the course is automatically become course-manager for this course.
3. The user is automatically being registered to the course.

Main success scenario:

1. The user clicks on the “Open course” bottom in his homepage.
2. The user provides the course details.
3. The system verifies the provided details.
4. The course is successfully created and becomes visible to students.
5. The user is automatically appointed as the course manager and being registered to the course.

Alternative flows**:**

1. **The course has already existed:** The system displays an error message stating the course is already in the platform.
2. **There is no such course in Ben Gurion University**: The system displays an error message stating that there is no such course being studied in Ben Gurion University.

### 2.2.5 UC 5 – Post an Exam

Description**:** A user wants to upload an exam to website.

Actor**:** User / Course Manager / System Manager

Parameters: exam file

Pre-conditions:

1. The user is registered and logged into the system.

Post-Conditions**:**

1. The exam is stored in the system database.

Main success scenario:

1. The user clicks on the “Upload exam” bottom in his homepage.
2. The user uploads the exam file.
3. The system automatically extracts the exam details – course name, year, semester (A/B/C) and moed (A/B/C).
4. The exam is being stored in the system database according to the details that haven extracted.
5. The exam is being listed and now can be seen in the course page on the website.

Alternative flows:

1. **The uploaded file format is invalid:** The system displays an error message and rejects the upload.
2. **The course which the exam belongs to, is not on the website**: The system displays an error message and suggest the option to open the course on the platform.
3. **The exam is already on the website:** The system displays an error message.
4. **The exam's details have not been extracted successfully**: The system displays an error message and send a notification to the course manager.

### 2.2.6 UC 6 – Post an Exam's Question

Description: A user posts a single question from a specific exam.

Actor: User / Course Manager / System Manager

Parameters: course name, year, semester, moed, the question file – pdf or photo.

Pre-conditions:

1. The user is registered and logged into the platform.

Post-Conditions:

1. The question is visible to other users on the platform.

Main success scenario:

1. The user clicks on the "post question" bottom in the homepage.
2. The user provides all the needed details of the question.
3. The user confirms and clicks "post".

4. The system confirms the question has been successfully posted.

Alternative flows:

1. **The course doesn't exist on the website:** The system displays an error message and suggest the option to open the course on the platform.

### 2.2.7 UC 7 – Search an Exams/Questions by Keywords or topics

Description: A user wants to find all the exams or the questions on the website that contains questions with those keywords or topics.

Actor: User / Course Manager / System Manager

Parameters: key words, topic

Pre-conditions:

1. The user is logged into the platform.

Post-Conditions:

1. A list of exams matching the keywords is displayed.

Main success scenario:

1. The user goes to the search bar at the homepage.
2. The user chooses the option of searching by keywords/topic.
3. The user enters keywords related to the exam they are looking for.
4. The system processes the keywords and displays a list of matching exams/ questions.

Alternative flows:

1. **No exams/questions match the keywords:** The system displays a message stating no results were found.

### 2.2.8 UC 8 – Search an Exam/Question by Specifics

Description: A user searches for a specific exam using specific details.

Actor: User / Course Manager / System Manager

Parameter: course name, year, semester, moed.

Pre-conditions:

1. The user is logged into the platform.

Post-Conditions:

1. The system navigates to the exam homepage.

Main success scenario:

1. The user goes to the search bar at the homepage.
2. The user chooses the option of searching by specific details.
3. The user applies the specific details.
4. The system displays the specific exam/question homepage.

Alternative flows:

1. **No exams/question match the filters:** The system displays a message stating no results were found.

### 2.2.9 UC 9 – Comment on a Question

Description**:** A user wants to ask about a specific question or share his thought and opinion about it. comments on a question by another user.

Actor: User / Course Manager / System Manager

Parameters: optional – photo or pdf file

Pre-conditions:

1. The user is logged into the platform.
2. The question exists in the system.

Post-Conditions:

1. The comment is visible under the question.

Main success scenario:

1. The user navigates to the question homepage.
2. The user presses “Add Comment” option.
3. The user writes / uploads file and submits the comment.
4. The system confirms the comment is added successfully.

Alternative flows:

1. **The comment is empty:** The system displays an error message prompting the user to add content to the comment.
2. **Violation of the platform policy**: The comment doesn't stand with the platform policy therefore will not be posted.

### 2.2.10 UC 10 – Comment on a Comment

Description: A user wants to add a comment to another user’s comment in a question’s homepage discussion.

Actor: User / Course Manager / System Manager

Parameters: optional – photo or pdf file

Pre-conditions:

1. The user is logged into the platform.
2. The comment exists in the system.

Post-Conditions:

1. The reply is visible under the original comment.

Main success scenario:

1. The user navigates to the question homepage.
2. The user presses “Add Comment” option in a specific comment in the question discussion.
3. The user writes / uploads file and submits the comment.
4. The system confirms the reply is added successfully.

Alternative flows:

1. **The reply is empty:** The system displays an error message prompting the user to add content to the reply.
2. **Violation of the platform policy**: The comment doesn't stand with the platform policy therefore will not be posted.

### 2.2.11 UC 11 – Appoint new course manager

Description: The Course Manager wishes to assign another user as a course manager for their course.

Actor: Course Manager/System Manager, user

Parameters: user’s email, course id

Pre-conditions:

1. The Course Manager is logged in.
2. The user exists in the system.
3. The course is assigned to the manager.

Post-Conditions:

1. The user is granted Course Manager privileges and responsibilities for the course.

Main Success Scenario:

1. The Course Manager navigates to the course settings.
2. The manager selects the user to appoint as a new course manager.
3. The user accepts the offer.
4. The system confirms the action and notifies the user.

Alternative Flows:

1. **The user declines**: The user doesn't wish to take the job and declines the offer.

### 2.2.12 UC 12 – Edit or remove course specific content

Description: The Course Manager wants to edit or remove questions, comments, or exams in their assigned course to ensure content quality and relevance.

Actor: Course Manager

Parameters: Content ID (question, comment, or exam), Action (Edit/Delete), updated content (for edits)

Pre-conditions:

1. The Course Manager is logged into the platform.
2. The content exists and is associated with the manager's assigned course.

Post-Conditions:

1. The content is updated or removed.

Main Success Scenario:

1. The Course Manager selects the content to edit or delete.
2. The manager performs the desired action.
3. The system confirms the action and reflects the changes.

Alternative Flows:

1. **The content is not found**: The system displays an error message.
2. **The user does not have permission for the action**: The system displays an error message.

#### 2.2.13 UC 13 – Remove Course Manager

Description: The System Manager revokes Course Manager privileges for a user.

Actor: System Manager

Parameters: User ID, Course ID

Pre-conditions:

1. The System Manager is logged in.
2. The user and course exist in the system.
3. The user is currently one of the course's managers.

Post-Conditions:

1. The user's Course Manager privileges are removed.

Main Success Scenario:

1. The Course Manager navigates to the course settings.
2. The manager removes the user from the managers list.
3. The system confirms the action and notifies the user.

Alternative Flows:

1. **The user or course does not exist**: The system displays an error message.
2. **The user is also a system manager**: The system informs the manager.

#### 2.2.14 UC 14 – Remove/Edit content

Description: The System Manager removes/edits inappropriate/wrong content flagged by users across the platform.

Actor: System Manager

Parameters: Content ID, Action (Edit/Delete/Ignore)

Pre-conditions:

1. The System Manager is logged in.
2. The flagged content exists in the system.

Post-Conditions:

1. The flagged content is updated, removed, or left unchanged.

Main Success Scenario:

1. The System Manager reviews flagged content in the moderation dashboard.
2. The manager decides on an action (edit/delete/ignore).
3. The system confirms the action and reflects the change.

Alternative Flows:

1. The flagged content is already addressed: The system notifies the manager

## 2.3. Special usage considerations

The NegevNerds platform should maintain high performance, especially during peak traffic periods, such as before exams. In case of system interruptions or crashes, the platform should recover seamlessly, preserving all previously saved data.

Additionally, if the integration with third-party tools, such as Google authentication, fails, the system should prevent the user from completing the registration process. However, this should not affect any other system functionalities, ensuring that the platform remains stable and operational for other users.

# 3. Functional requirements

## 3.1. System requirements

3.1.1. **User courses:** The system will display the courses the user is registered to on the homepage.

3.1.2. **Course exams:** The system will display all the exams associated with a course on its page.

3.1.3. **Search results:** The system will display the search query results.

3.1.4. **Display search results by topic:** The system will display search results in descending order of relevance, prioritizing questions with more frequent keywords.

3.1.5. **Auto tagging:** The system will automatically tag questions based on the topics in the course syllabus.

3.1.6. **Edited comments:** The system will indicate when a comment in a discussion has been deleted or edited.

3.1.7. **Teacher solution:** The system will display the teacher's solution to the question, if available.

3.1.8. **User reports:** The system will forward user reports to the course manager. If there is no course manager, the report will be sent to the system manager.

3.1.9. **Real Time notifications:** The system will support real-time notifications for users who are online on the website.

3.1.10. **Delayed notifications:** The system will support delayed notifications for users who were offline.

3.1.11. **Syllabus:** While creating new course, the system will require to upload it's syllabus.

## 3.2. User Requirements

3.2.1. <u>Guest – not registered</u>

3.2.1.1. **Register:** A guest who is not registered can register by providing unique and valid identifying details, including a BGU email and password. After


successfully completing the registration process, the user status is logged in.

3.2.2. Guest – registered

3.2.2.1. **Log in:** A registered guest can log in using his email and password.

3.2.2.2. **Password recovery:** A guest can recover his password using the email he is registered with.

3.2.3. User

3.2.3.1. **Course registration:** User can register to courses.

3.2.3.2. **Open new course:** User can create a new course.

3.2.3.3. **Upload exam:** User can upload a new complete exam to the system.

3.2.3.4. **Upload question:** User can upload a new single question to the system, where the exam it belongs to does not exist on the site. In this case, they will need to manually enter the question details.

3.2.3.5. **Discussion comment:** User can comment on a discussion using free text or by uploading a file.

3.2.3.6. **Comment on comment:** User can comment on a discussion comment.

3.2.3.7. **React comment:** User can react to comments by using informative emojis (e.g. “like”, “plus”, “thanks”, etc).

3.2.3.8. **Search questions:** User can search for exam questions in two different ways:

- By specific date: search for a specific question from a certain exam.
- By keywords: search for questions on a specific topic by entering keywords in the search bar.

3.2.3.9. **Search exams:** User can search for exam questions in two different ways:

- By specific date: search for a specific exam.
- By course: search for exams of a specific course.

3.2.3.10. **Log out:** User can log out of the system.


3.2.3.11. **Report inappropriate content:** User can report about inappropriate content or content that violates community guidelines to system managers.

3.2.3.12. **Report issues with questions:** User can report inconsistencies between a question and its answer/date.

3.2.3.13. **Profile management:** User can edit his personal profile details.

3.2.3.14. **Tag question:** User can add a tag to a question.

3.2.3.15. **Manage answers:** User can delete or edit his comments.

3.2.3.16. **Following discussion:** User can follow discussion.

3.2.3.17. **Receive notifications:** User can choose to receive notifications in the following cases:

- A new exam is uploaded to a course he is registered to.
- Any new comment on a course he is registered to.
- A new comment in a discussion he was participated in.
- A new comment in a discussion he is following.
- A comment/reaction received on a comment he posted.
- Updates from the system manager.
- Updates from the course manager of courses he is registered to.

3.2.4. User - System Manager

3.2.4.1. **Assign course manager:** A system manager can assign a new course manager for a specific course from the list of existing users.

3.2.4.2. **Remove course manager:** A system manager can remove a course manager from his role.

3.2.4.3. **Delete question:** A system manager can delete irrelevant questions or those with wrong tags.

3.2.4.4. **Edit question:** The system manager can edit the details of a question (e.g., year, semester, question number, tags).

3.2.4.5. **Delete exam:** A system manager can delete an exam that has been published and does not comply with the system's policy.

3.2.4.6. **Edit exam:** A system manager can edit the details of an exam (e.g., year, semester).

3.2.4.7. **Delete comment:** A system manager can delete inappropriate or incorrect comments.

3.2.4.8. **Self-removal:** In case there are more system managers, a system manager can remove himself from the role.

3.2.4.9. **Notifications:** A system manager will receive reports about issues with questions or discussions.

3.2.5. User - Course Manager

3.2.5.1. **Delete question:** A course manager can delete irrelevant questions or those with wrong tags related to his course.

3.2.5.2. **Edit question:** A course manager can edit the details of a question (e.g., year, semester, question number, tags) related to his course.

3.2.5.3. **Delete exam:** A course manager can delete an exam related to his course that has been published and does not comply with the system's policy.

3.2.5.4. **Edit exam:** A course manager can edit the details of the exam (e.g., year, semester) related to his course.

3.2.5.5. **Delete comment:** A course manager can delete inappropriate or incorrect comments related to his course.

3.2.5.6. **Self-removal:** A course manager can remove himself from the role.

3.2.5.7. **Notifications:** A course manager will receive reports about issues with questions or discussions related to his course.

# 4. Non-functional requirements

## 4.1. Implementation constraints

### 4.1.1 - Speed requirements

**Registration**: The system should process registration requests within no longer than 30 seconds.

**Log in:** The system should process log-In to the system within no longer than 5 seconds.

**Search by date:** The system should display the search results within 5 seconds.

**Search by key words:** The system should display the search results within 15 seconds.

**Question upload:** The system should finish the upload of specific question within 30 seconds.

**Exam upload:** The system should finish the upload of a full exam within 1 minute.

**Post a comment:** Post a comment to discussion should take no more than 5 seconds.

**Push notifications:** The system will be sent within **3 second** of a triggered event.

### 4.1.2- Capacity requirements

The system will support **up to 1,000 concurrent users** during normal operation, with the ability to scale to **5,000 concurrent users** during peak periods, such as before exams.

The system will limit individual file uploads to **10 MB per file** to ensure consistent performance.

A single user can upload up to **3 files per minute**, with the server queueing excess uploads during heavy traffic.

The total size of all media files stored in the database must not exceed **32 GB** at any given time.

### 4.1.3 – Safety and Security

User passwords and personal data should be stored using **encryption standards: AES-256**.

### 4.1.4 – Portability

The system will be accessible via any modern web browser (Chrome, Firefox, Edge) and on both desktop and mobile devices.

### 4.1.5 - Reliability requirements

In case of a server failure, the system should recover and return to full functionality within **10 minutes**.

Data integrity will be maintained for all user-generated content, even during unexpected shutdowns or crashes.

### 4.1.6 – Usability

The system should follow user-friendly design principles, making it accessible to users with minimal technical skills.

Navigation should be intuitive, with actions such as creating, searching, and responding to discussions requiring no more than three clicks.

Error messages should be clear and provide actionable steps for resolution (e.g., “Your session expired. Please log in again.”).

### 4.1.7 – Availability

The system should be operational and accessible 24/7, except during scheduled maintenance windows.

The system will ensure that offline backups are conducted daily, and real-time database replication is in place to prevent data loss.

## 4.2. Platform constraints

The backend of the system will be developed in **Python**, utilizing the Flask framework for server-side logic and database management.

The frontend will be developed using **React**, enabling a modern and responsive user interface accessible through web browsers.

The website will be hosted on Apache as the web server

The database will be managed using **MySQL**, chosen for its reliability and open-source availability.

The system must run on modern web browsers such as **Google Chrome**, **Mozilla Firefox**, and **Microsoft Edge**, ensuring compatibility across widely used platforms.

The system will use open-source libraries such as **Apache Lucene 2.0** and **Tesseract OCR** to enhance text processing and recognition capabilities, facilitating advanced search functionalities and document handling.

### 4.2.1. SE Project constraints

**System Interactivity**: The system is interactive and designed for collaborative use by students preparing for exams. Users can post and respond to exam-related questions.

**Input Source**: Inputs will originate directly from users interacting with the system via their web browsers, entering text or uploading files.

**Data Used in Testing**: Testing will utilize a combination of randomly generated data and real-world samples, such as mock exam questions and answers contributed by participants.

**Hardware Requirements**: The system does not require specialized hardware. It will run on standard hardware configurations, ensuring accessibility during development and testing phases.

**Access Rights and Security**: The system's functionality is restricted to registered users – BGU students and employees. During testing, controlled user accounts with varying access rights will be created to validate system security.

**Development and Testing**: Development will occur on commonly used development environments. Testing will involve both functional tests and concurrency tests to ensure the system can handle multiple users simultaneously.

**Presentation of Final System**: The system will be demonstrated during the course's project presentation day by:

1. Presenting data collected in the alpha and beta versions, including exam-related content uploaded by users in previous versions.
2. Allowing participants to log in and interact with the forum interface.
3. Showing users posting questions, searching for answers, and responding to existing questions.
4. Simulating scenarios where users attempt unauthorized actions (e.g., posting without logging in) to demonstrate the system's access control mechanisms.
5. Using an administrator account to manage content and user privileges.
6. Demonstrating the system's responsiveness under simultaneous user activity.

## 4.3. Special restrictions & limitations

The following assumptions and special restrictions influence the design and implementation of the system:

**Copyrights:**

The system assumes that only registered users (BGU students and employees) can access the system.

**User Disabilities:**

The system does not currently account for users with specific disabilities (e.g., visual impairments, motor disabilities).

**Appropriate language:**

Students and system managers must maintain appropriate language. Enforcement of this requirement is the responsibility of the managers.

## 5. Risk assessment & Plan for the proof of concept

**Objectives of the Prototype:**

The primary goal of the proof-of-concept for the NegevNerds system is to validate the feasibility of the system both technically and functionally, while minimizing risks in later development stages. This prototype will focus on testing key features of the system, such as the user management, questions search engine (by date only for this version) and post questions or exams.

**Search Functionality:**

- Verify the ability to search for questions by date.
- Measure the performance of the search engine with a small dataset to understand its limitations.

**Post questions:**

- Develop an initial forum system where users can post questions and full exams.

**User Management:**

- Test the registration and login mechanism using BGU email addresses.
- Ensure role-based access control between regular users and system managers.

**Benefits of the Prototype for the Project:**

**Better Understanding of Requirements:**

- By developing preliminary components of the system, we can gain a deeper understanding of the needs of end-users and potential technical requirements.

**Evaluation of Technological Capabilities:**

- The prototype will allow us to test the performance and capabilities of the chosen technologies, such as React for the frontend and Python with Flask for the backend.

**Assessment of Design Decisions:**

- We will evaluate whether the modular structure of the system allows for flexibility and future scalability.
- Based on test results, we can modify components early and avoid design flaws in later stages.

**Visualization for the Customer:**

- Through the prototype, users will visualize how the final system will look and function.
- The demonstration will provide valuable feedback to improve the overall user experience.

**Risk Reduction:**

- Early identification of potential issues in the search engine, forum performance, or system load handling.
- Establishing a gradual process where problems are addressed before advancing to full development.

**Operational Knowledge:**

- The prototype offers valuable experience in launching a live website, including deploying the platform on a server or hosting environment.
- This stage also involves hands-on learning in building and structuring a functional and scalable database, a skill critical for future development.

**Prototype Stages and Demonstration:**

1. Build a small database with example questions, responses, and ratings.
2. Develop a basic search engine and test its efficiency under various scenarios.
3. Create a simple user interface with pages for registration to the system, login, open new courses, register for existing courses, search, and post questions or exams.
4. Present the prototype to testers and users for feedback collection.

# 6. Appendices

## 6.1. Input and Output Formats

**Input Formats:**

The system accepts exams uploaded in the following format:

- Scanned exams must be clear and readable.
- Supported fonts and layouts: Exams with structured questions and clear separations (e.g., "Question 1").

Example PDF with a typical structure on the next page (39-43).

**Output Formats:**

The system processes the input and provides the following outputs:

1. Separate PDF Files for Questions:
   - Each question is saved as an individual PDF file.
   - File naming convention: question_1.pdf, question_2.pdf, ..., question_n.pdf. (example for split pdf in page 44-45)
2. Metadata Extraction:
   - Course Name (e.g., "Operating Systems")
   - Course Number (e.g., "20213031")
   - Year (e.g., "2019")
   - Semester (e.g., "B")
   - Exam Session (e.g., "Moed B")

**אוניברסיטת בן-גוריון בנגב, המחלקה למדעי המחשב**

# ***מועד ב' במערכות הפעלה***

| |
|---|
| תאריך הבחינה: 15 ביולי, 2019<br>שם הקורס: מערכות הפעלה<br>מספר הקורס: 20213031<br>שנה: 2019, סמסטר: ב', מועד: ב'<br>משך הבחינה: שלוש שעות<br>חומר עזר: אסור |

**מרצים: איתי דינור, דני הנדלר ומרינה קוגן-סדצקי.**

**מתרגלים: אור דינרי, מתן דרורי, סימיון נוביקוב, צחי ספורטה, עמית פורטנוי וירין קופר.**

**ענו על כל השאלות: סה"כ 100 נקודות.**

**1. ניהול זיכרון (25 נקודות)**

א. **(16 נקודות)** בסעיף זה נתייחס למערכת הפעלה התומכת בזיכרון וירטואלי המנוהל במנגנון paging. גודל דף במערכת הוא 8 KB. המערכת תומכת בשתי רמות של טבלאות דפים, כאשר ברמה הראשונה יש טבלה אחת. גודל כל טבלת דפים (בכל רמה שהיא) הוא דף אחד (כלומר 8 KB) וגודל כניסה בכל טבלת דפים הוא 4 בתים. הקצאות הזיכרון במערכת הן תמיד בכפולות של 8 KB.

ענו על השאלות הבאות ונמקו את תשובותיכם בקצרה.

i. **(6 נקודות)** מהו הגודל המקסימלי של הזיכרון הווירטואלי שהמערכת יכולה לתמוך בו?

ii. **(6 נקודות)** נתון תהליך משתמש אשר מנצל 68 MB (68 מגה-בית) ממרחב הזיכרון הווירטואלי שלו וכולם נמצאים בזיכרון הפיסי. מהו המספר המינימאלי של טבלאות דפים הדרוש לתהליך (בשתי הרמות)?

iii. **(4 נקודות)** בהמשך ל-ii, נתון שמערכת ההפעלה ביצעה swap out לדיסק ל 32 MB של זיכרון מתוך ה 68 MB של התהליך. ענו על אותה שאלה לאחר ביצוע ה swap out.

ב. **(9 נקודות)** נתונה מערכת הפעלה התומכת בזיכרון וירטואלי המנוהל במנגנון paging. בנוסף נתון כי מערכת ההפעלה רצה על מחשב עם מעבד יחיד ורשומות ה TLB במעבד אינן מכילות שדה של מזהה תהליך. נאמר שפעולה מסוימת מחייבת שינוי של ה TLB, אם קיים תרחיש שבו ביצוע הפעולה ללא שינוי ה TLB גורר פגיעה בנכונות תוכנית משתמש. עבור כל אחת מהפעולות מבין הפעולות הבאות, רשמו האם היא מחייבת שינוי של ה TLB עבור המערכת הנתונה. נמקו את תשובותיכם בקצרה.

i. **(5 נקודות)** מיפוי (הקצאה) של דף חדש בזיכרון הווירטואלי של התהליך הנוכחי (התהליך הרץ על המעבד) לדף בזיכרון הפיסי.

ii. **(4 נקודות)** ביצוע swap out לדיסק עבור דף הממופה לזיכרון הווירטואלי של התהליך הנוכחי.

**2. חוטים ותהליכים (25 נקודות)**

נתונה התוכנית הבאה. תכנית זו יוצרת שני חוטים. החוט הראשון מבצע את הפונקציה f1 והחוט השני מבצע את הפונקציה f2. החוט הראשי ממתין לסיום של שני החוטים הללו לפני שהוא מבצע את שתי פקודות ההדפסה. לאורך כל השאלה יש להניח שכל הקריאות ל-thread_create, thread_join ו-fork מצליחות. **שימו לב: הסעיפים הבאים אינם קשורים זה לזה. כל השינויים בסעיפים אלו מתייחסים לתוכנית המקורית המופיעה להלן.**

```
// Global variables
int g1 = 0, g2 = 0;

f1( )
{ g2 = g1 + 1; }

f2( )
{ g1 = g2 – 1; }

main( )
{
    thread_create(f1);
    thread_create(f2);

    // Wait for all threads to terminate
    while (thread_join( ) == 0);

    printf("g1 = %d", g1);
    printf("g2 = %d", g2);
}
```

**א. (6 נק')** מה הם הפלטים האפשריים של התוכנית? עבור כל פלט אפשרי, כתבו מהו והסבירו בקצרה.

**ב. (6 נק')** הוחלט להחליף את הפונקציה main בפונקציה הבאה. כיצד ישתנה הפלט של התוכנית (אם בכלל)? הסבירו בקצרה, התייחסו לכל התוצאות האפשריות.

```
main( )
{
    if (fork() == 0)
        {
        f1();
        return;
        }
    if (fork() == 0)
        {
        f2();
        return;
        }
    while (wait( ) == -1); // Wait for all child processes to terminate
    printf("g1 = %d", g1);
    printf("g2 = %d", g2);
}
```


ג. **(6 נק')** הוחלט לשנות את התוכנית המקורית ע"י הוספת שני משתנים גלובליים מסוג **mutex** ולשנות את הפונקציות f2 ו-f2 כלהלן. כיצד ישתנה הפלט של התוכנית (אם בכלל)? הסבירו בקצרה, התייחסו לכל התוצאות האפשריות.

```
mutex m1, m2 // Two additional global variables
f1( )
{
        lock(m1);
        g2 = g1 + 1;
        unlock(m1);
}
f2( )
{
        lock(m2);
        g1 = g2 – 1;
        unlock(m2);
}
```

ד. **(7 נק')** הוחלט לשנות את התוכנית המקורית ע"י הוספת שני משתנים גלובליים מסוג **mutex** ולשנות את הפונקציות f2 ו-f2 כלהלן. כיצד ישתנה הפלט של התוכנית (אם בכלל)? הסבירו בקצרה, התייחסו לכל התוצאות האפשריות.

```
mutex m1, m2 // Two additional global variables
f1( )
{
        lock(m1);
        lock(m2);
        g2 = g1 + 1;
        unlock(m2);
        unlock(m1);
}
f2( )
{
        lock(m2);
        lock(m1);
        g1 = g2 – 1;
        unlock(m1);
        unlock(m2);
}
```


**3. סינכרוניזציה (25 נקודות)**

**א. (12 נק')** להלן קוד המממש סמאפור בינארי באמצעות מוניטור.

```
BinarySemphore: Monitor {

1 condition variable cv
2 int free initially 1
3 void down( ) {
4    while (free ≤ 0)
5       { cv.wait }
6    free--
6 }
7 void up( ) {
8    if (free < 1) {
9       free++
10      cv.signal
11   }
12 }
```

i. **(6 נק')** בהינתן שהמוניטור הוא מסוג Hoare, האם המימוש הנ"ל מקיים את הסמנטיקה של סמאפור בינארי? אם כן הסבירו בקצרה, אחרת תארו תסריט מדויק המהווה דוגמא נגדית.

ii. **(6 נק')** ענו על אותה שאלה בהינתן שהמוניטור אינו מסוג Hoare.

**ב. (13 נק')** להלן אלגוריתם למניעה הדדית.

```
shared int turn
shared boolean busy initially false
Program for process i ∈ (1,...,n)
1 do {
2     do {
3         turn := i
4         } while (busy)
5     busy := true
6     } while (turn != i)
7 Critical section
8 busy := false
```

i. **(6 נק')** האם האלגוריתם מקיים מניעה הדדית? אם כן, נמקו בקצרה. אם לא, כתבו תסריט מדויק המדגים כי יש הפרה של מניעה הדדית.

ii. **(7 נק')** האם האלגוריתם מקיים חופש מקיפאון? אם כן, נמקו בקצרה. אם לא, כתבו תסריט מדויק המדגים כי אין חופש מקיפאון.

**4. ניהול קבצים, וירטואליזציה (25 נקודות)**

בכל הסעיפים בשאלה זו, יש לנמק במדויק ובקצרה.

**א. (5 נק')** במערכת הפעלה UNIX נמחק התוכן של קובץ הספרייה /usr שהכילה קבצים רגילים בלבד. ידוע שבספריות אחרות על הדיסק, קיים לכל אחד מקבצים אלה גם soft link וגם hard link. האם ניתן על סמך links אלו לשחזר את תוכנו של קובץ הספרייה?

**ב.** נניח כי בזמן נתון, במערכת הפעלה UNIX, קיים קובץ /file וקיימים X inodes פנויים.

i. **(3 נק')** זרובבל יוצר hard link חדש לקובץ /file, כמה inodes פנויים יש במערכת כעת?

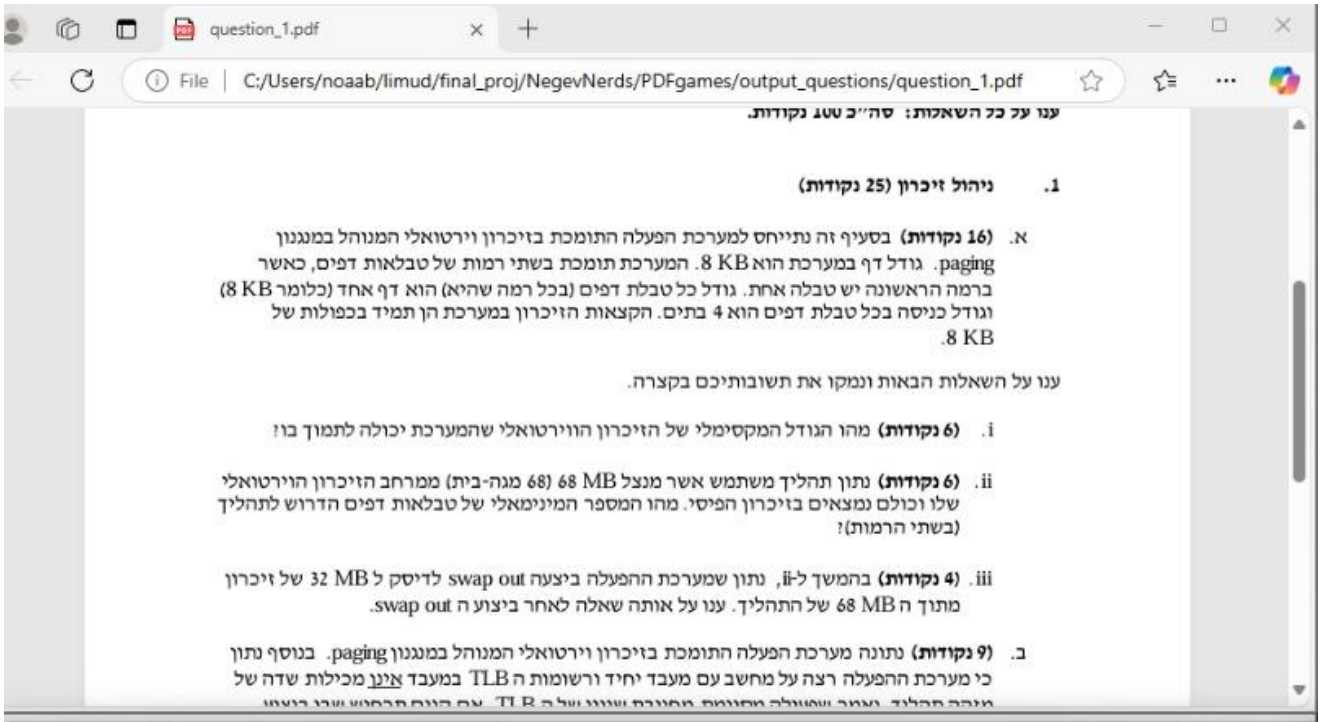
ii. **(3 נק')** ענו על אותה שאלה בהנחה שזרובבל יוצר soft link חדש.

**ג. (5 נק')** מהו קובץ דליל (sparse file)?

**ד. (4 נק')** הסבירו בקצרה מהי privileged instruction ומהי sensitive instruction.

**ה. (5 נק')** הסבירו בקצרה מהי וירטואליזציה בשיטת trap and emulate ומדוע היא לא הייתה אפשרית בארכיטקטורות ישנות של x86, בהן לא הייתה תמיכה של החומרה בוירטואליזציה.

***בהצלחה!***


question_1.pdf

File | C:/Users/noaab/limud/final_proj/NegevNerds/PDFgames/output_questions/question_1.pdf

ענו על כל השאלות: סה"כ 100 נקודות.

**1. ניהול זיכרון (25 נקודות)**

א. **(16 נקודות)** בסעיף זה נתייחס למערכת הפעלה התומכת בזיכרון וירטואלי המנוהל במנגנון paging. גודל דף במערכת הוא 8 KB. המערכת תומכת בשתי רמות של טבלאות דפים, כאשר ברמה הראשונה יש טבלה אחת. גודל כל טבלת דפים (בכל רמה שהיא) הוא דף אחד (כלומר 8 KB) וגודל כניסה בכל טבלת דפים הוא 4 בתים. הקצאות הזיכרון במערכת הן תמיד בכפולות של 8 KB.

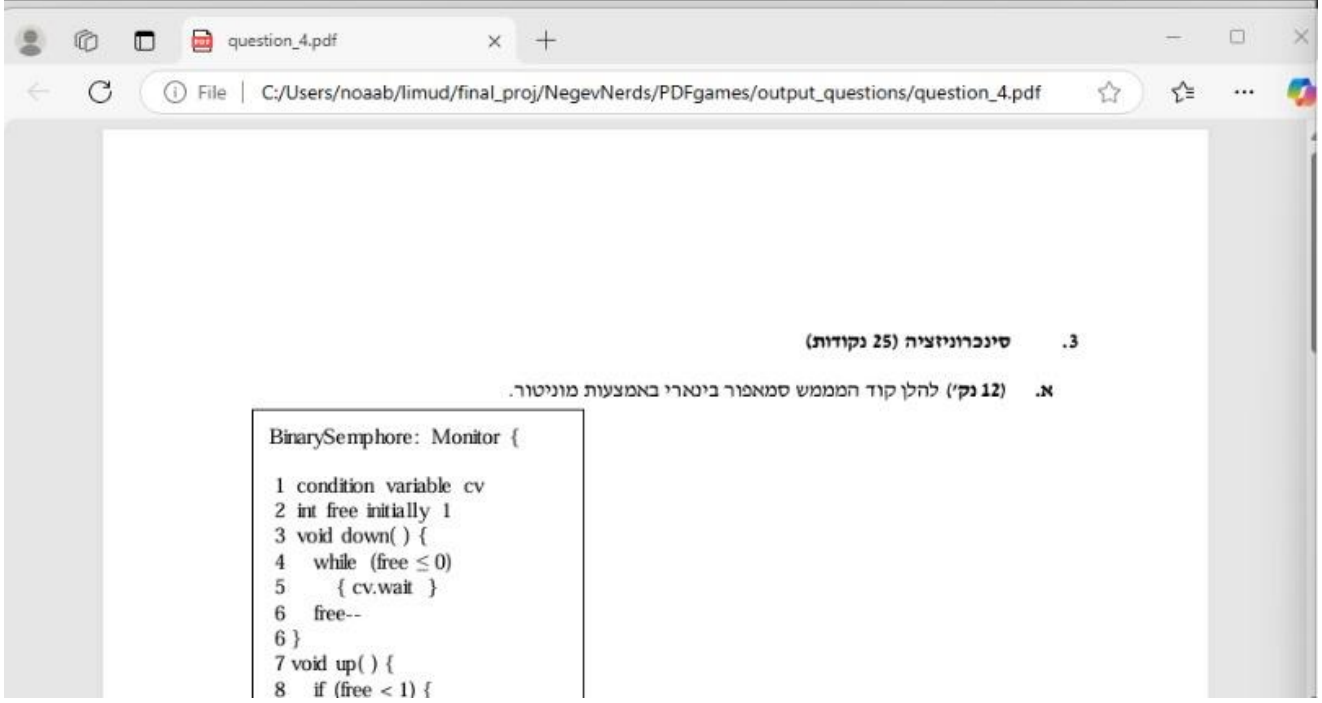
ענו על השאלות הבאות ונמקו את תשובותיכם בקצרה.

i. **(6 נקודות)** מהו הגודל המקסימלי של הזיכרון הווירטואלי שהמערכת יכולה לתמוך בו?

ii. **(6 נקודות)** נתון תהליך משתמש אשר מנצל 68 MB (68 מגה-בית) ממרחב הזיכרון הוירטואלי שלו וכולם נמצאים בזיכרון הפיסי. מהו המספר המינימאלי של טבלאות דפים הדרוש לתהליך (בשתי הרמות)?

iii. **(4 נקודות)** בהמשך ל-ii, נתון שמערכת ההפעלה ביצעה swap out לדיסק ל 32 MB של זיכרון מתוך ה 68 MB של התהליך. ענו על אותה שאלה לאחר ביצוע ה swap out.

ב. **(9 נקודות)** נתונה מערכת הפעלה התומכת בזיכרון וירטואלי המנוהל במנגנון paging. בנוסף נתון כי מערכת ההפעלה רצה על מחשב עם מעבד יחיד ורשומות ה TLB במעבד אינן מכילות שדה של


question_4.pdf

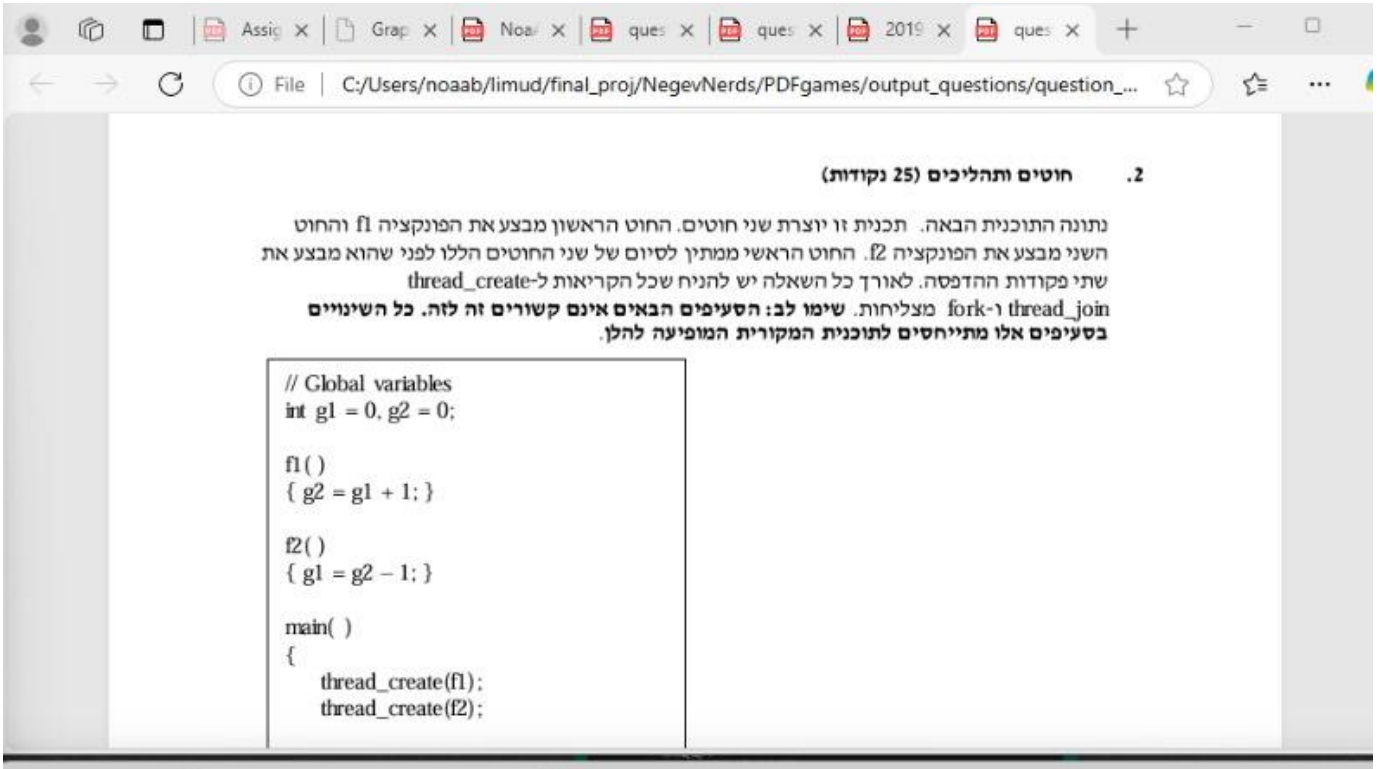
File | C:/Users/noaab/limud/final_proj/NegevNerds/PDFgames/output_questions/question_4.pdf

**3. סינכרוניזציה (25 נקודות)**

א. **(12 נק')** להלן קוד המממש סמאפור בינארי באמצעות מוניטור.

```
BinarySemphore: Monitor {

1 condition variable cv
2 int free initially 1
3 void down( ) {
4    while (free ≤ 0)
5       { cv.wait }
6    free--
6 }
7 void up( ) {
8    if (free < 1) {
```




**2. חוטים ותהליכים (25 נקודות)**

נתונה התוכנית הבאה. תכנית זו יוצרת שני חוטים. החוט הראשון מבצע את הפונקציה f1 והחוט השני מבצע את הפונקציה f2. החוט הראשי ממתין לסיום של שני החוטים הללו לפני שהוא מבצע את שתי פקודות ההדפסה. לאורך כל השאלה יש להניח שכל הקריאות ל-thread_create ו-thread_join ו-fork מצליחות. **שימו לב: הסעיפים הבאים אינם קשורים זה לזה. כל השינויים בסעיפים אלו מתייחסים לתוכנית המקורית המופיעה להלן.**

```
// Global variables
int g1 = 0, g2 = 0;

f1( )
{ g2 = g1 + 1; }

f2( )
{ g1 = g2 – 1; }

main( )
{
    thread_create(f1);
    thread_create(f2);
```


**4. ניהול קבצים, וירטואליזציה (25 נקודות)**

בכל הסעיפים בשאלה זו, יש לנמק במדויק ובקצרה.

**א. (5 נק')** במערכת הפעלה UNIX נמחק התוכן של קובץ הספרייה usr/ שהכילה קבצים רגילים בלבד. ידוע שבספריות אחרות על הדיסק, קיים לכל אחד מקבצים אלה גם soft link וגם hard link. האם ניתן על סמך links אלו לשחזר את תוכנו של קובץ הספרייה?

**ב.** נניח כי בזמן נתון, במערכת הפעלה UNIX, קיים קובץ file/ וקיימים X inodes פנויים.

i. **(3 נק')** זרובבל יוצר hard link חדש לקובץ file/ , כמה inodes פנויים יש במערכת כעת?

ii. **(3 נק')** ענו על אותה שאלה בהנחה שזרובבל יוצר soft link חדש.

**ג. (5 נק')** מהו קובץ דליל (sparse file)?

**ד. (4 נק')** הסבירו בקצרה מהי privileged instruction ומהי sensitive instruction.

## 6.2. Similar product analysis

1. **WhatsApp Groups**:

- **Functionality**:
    - Students use WhatsApp to share past exam questions and discuss answers in real-time.
    - Offers a quick way to receive answers from peers.
- **Strengths**:
    - Instant communication.
    - Widely used and accessible to all students.
- **Weaknesses**:
    - Conversations are hard to revisit and organize.
    - No search functionality for specific topics or keywords.
    - Reliance on peers from the same academic year.
    - Easily disrupted by unrelated discussions.

2. **Google Drive**:

- **Functionality**:
    - Students upload and share past exams, solutions, and lecture notes.
- **Strengths**:
    - Centralized storage of documents.
    - Easy to use and access.
- **Weaknesses**:

- Lacks search functionality for specific questions or topics.
- No tools for discussion or collaboration.
- Information is not structured or organized effectively.

3. **Moodle**:

- **Functionality**:
  - Some courses provide forums or repositories for exam questions and discussions.
- **Strengths**:
  - Managed by the course team, ensuring quality and reliability.
  - Formal structure for asking and answering questions.
- **Weaknesses**:
  - Answers are not immediate.
  - Search capabilities for specific questions or topics are limited.
  - Dependent on the course team's engagement.