

**HACETTEPE UNIVERSITY DEPARTMENT OF COMPUTER ENGINEERING**  
**BBM384 SOFTWARE ENGINEERING LABORATORY**  
**EXPERIMENT SHEET**  
**2021 Spring**

Instructors: Assoc. Prof. Dr. Ayça KOLUKISA TARHAN, Asst. Prof. Dr. Ebru GÖKALP, Dr.  
Tuğba Gürgen ERDOĞAN

Teaching Assistants: R. A. Burcu YALÇINER, R.A. Bahar GEZİCİ

## 1. Purpose and Scope

The purpose of this experiment sheet is to guide students who enroll BBM384 course in developing their laboratory projects. The projects are pre-defined and the process defined in this document is applied through the development of the projects. While applying the process, the students follow the schedule given by the instructors at the beginning of the course.

## 2. Abbreviations

SDLC	:	Software Development Life Cycle
SRS	:	Software Requirements Specification
TCD	:	Test Case Definition
SDD	:	Software Design Description
STR	:	Software Test Report
GUI	:	Graphical User Interface

## 3. Roles and Responsibilities

<b>Roles</b>	<b>Responsibilities</b>
Student	Responsible for conforming the requirements of this process, and for timely generation of the outputs of this process.
Teaching Assistant(s)	Responsible for timely evaluation and feedback of the outputs generated by the students.
Instructor(s)	Responsible for resolution of issues likely to occur between teaching assistants and the students.
Stakeholder	Any person that has a relation with the system to be developed (including students, teaching assistants, and instructors).
Team Member	A student that takes role in the development of the system. Undertaken roles include Software Project Manager, Software Analyst, Software Architect, Software Developer, Software Change/Configuration Manager, and Software Tester. Each student must have <u>Software Developer</u> role during development. A student may have more than one role during development.

#### 4. Inputs

- (i) Vision Document Template
- (ii) Project Plan Template
- (iii) Software Requirements Specification Template
- (iv) Use Case Definition Template
- (v) Test Case Definition Template
- (vi) Graphical User Interface Design Template
- (vii) Architecture Notebook Template
- (viii) Risk Management Report Template
- (ix) Configuration/Change Management Report Template
- (x) Software Design Description Template
- (xi) Software Coding Standard
- (xii) Test Script Template
- (xiii) Software Test Report Template
- (xiv) Github Flow: <https://guides.github.com/introduction/flow/>

#### **NOTE:**

- Risk Management Report will be created by using the lists on the page:  
<https://docs.google.com/spreadsheets/d/1jhju-n0OYEwb8HPvUaL2W-027xRs-Sw/edit#gid=768757863>
- Configuration/Change Management Report will be created by using the lists on the page:  
<https://docs.google.com/spreadsheets/d/1jhju-n0OYEwb8HPvUaL2W-027xRs-Sw/edit#gid=9565548>

#### 5. Outputs

- (i) Software Vision Document
- (ii) Software Project Plan
- (iii) Software Requirements Specification (SRS)
- (iv) Use Case Definitions (as an attachment to SRS)
- (v) Graphical User Interface Definitions (as an attachment to SRS)
- (vi) Test Case Definitions (as an attachment to SRS)
- (vii) Architecture Notebook
- (viii) Risk Management Report
- (ix) Configuration/Change Management Report
- (x) Software Design Document
- (xi) Software Code (and Software Coding Standard if modified)
- (xii) Software Test Report (STR)
- (xiii) Test Case and Test Script Definitions (as an attachment to STR)
- (xiv) Project's Github Repository (master and member branches, issues, pull requests, commits etc.)

#### 6. System To Be Developed

You are expected to develop an online system for students (Bachelor's students, Master students, PhD students and Exchange Students), graduates and academicians at Hacettepe University Computer Engineering Department. This system will be similar in functionality to LinkedIn system ([www.linkedin.com](http://www.linkedin.com)). The name of our system is **LinkedHU\_CENG**. This system will be ideal system for students, graduates and academicians at Hacettepe University Computer Engineering Department looking to connect with each other.

LinkedHU\_CENG system will be such platform that enables users to: enroll in the system; create their own profiles; share scholarship, job, or any other opportunities related to the industry; share and watch presentations; enjoy events such as sport, conference, dinner, music etc. according to their interests and hobbies; join online meetings; chat with the person who wants to connect with; and track the news related to new job opportunities, events, online meetings, videos etc.

While developing this system, you are expected to develop a flexible, maintainable, attractive, and extensible system. Your application should provide Open-Close Principle (i.e. it should be closed for modification and open for extension).

All software development activities will be done as a GitHub project. GitHub is the best way to build software together. Each member must create a branch where s/he can safely experiment and make changes by the following [GitHub flow](#). S/he will use a pull request to get feedback on his/her changes. After the changes are reviewed, s/he will merge them into the master branch and deploy his/her code to the project.

**System Modules:** The system named LinkedHU\_CENG includes four main modules. These are: administration module, student module, graduate module, and academician module.

#### ***Administration Module:***

- Logins to the system by entering his/her login id and password.
- Manages his/her account and member accounts. For example, s/he can send new password to the user who forgot his/her password.
- Merges a user's e-mails to a specific e-mail address and adds a new user's e-mail into it.
- Gets user details, which are personal information (name, surname, phone number, e-mail address), company that s/he works for, and link to LinkedIn profile if exists.
- Downloads user details in Excel format and sends it to the user who wants to access this Excel file.
- Manages videos uploaded by authorized users. For example, s/he should delete the video whose context is not appropriate.
- Manages news and announcements uploaded by authorized users.
- Takes new enrollment requests sent from new users and approves his/her request.
- Deletes the students who are dismissed from the university and the academicians resigned from his/her faculty member profession.
- Manages the schedules of online meetings and events when there is a conflict.
- If any user uses slang words in messages or during the online/offline events & meetings, or there is a complaint about him/her, the admin should ban this user. The banned user cannot send any messages to other users and cannot join any online/offline events & meetings.
- If the student who has a specific role as "student representative" uses slang words in messages or during the online/offline events & meetings, or there is a complaint about him/her, the admin should ban this user. Admin can impose sanctions to the student representative mentioned in the previous sentence and in addition to these sanctions, the status of the banned student representative is changed from student representative to normal student. Therefore, s/he cannot be student representative again.

### ***Student Module:***

- There are three types of students: Bachelor's student, Master student, PhD student.
- There is also a student who has additional authorities per student type. This student is student representative of all students in that student type.
- Students can chat with graduates, academicians and other students.
- Students can enroll to the system and send enrollment requests to the admin.
- Students can login to the system by entering his/her login id and password.
- Students can search & view faculty presentations in the video library and attend live presentations on event day.
- Students can write questions and comments under online/offline presentations.
- Students can view announcements related to specific online/offline presentations.
- Students can access/manage their accounts, and can change their personal information such as password.
- Students can attend live events & online meetings.
- Students can also view the announcements & news about live events & online meetings on their main page. By clicking on announcements & news, they can see the details of them.
- Students can view scholarships, internships and job advertisements, and they can apply them by entering necessary information and uploading specific documents (e.g. resume and certificates). The only permitted format for resumes is .pdf and for certificates are .pdf, .jpeg and .png. The system should give error messages when users try to upload resumes and certificates with different formats and when they do not fill the necessary information and/or write wrong information.
- Students can request for lecture notes, lecture videos, articles, conference papers and any other materials they need from academicians.
- Student representative can add new event to the system and send e-mails to all users of the system by using e-mail addresses in bulk.
- Student representative can add new announcement to the system.
- Student representative can manage the schedule of live events & online meetings when the academician and graduate who arrange the event want him/her make any change.
- Student representative can take requests and suggestions from other students and forward them to the academicians, graduates and admin.
- Student representative can request the all users' information from the admin.

**NOTE:** The additional functionalities that the student representative role has will be enabled in the same module and these functionalities for other students will be disabled.

### ***Academicians Module:***

- Academicians can chat with students, graduates and other academicians.
- S/he can enroll to the system and send enrollment request to the admin.
- S/he can add internship announcements to the system.
- S/he can add scholarship announcements to the system.
- S/he can make online/offline presentations.
- S/he can write questions and comments under online/offline presentations.
- S/he can make announcement for the research assistant recruitment.
- S/he can also make announcements for the recruitment of faculty members.
- S/he can create live events & online meetings.
  - Live events and online meeting should be scheduled by determining the user's availability via some online tools like when2meet.

- Online meeting should be done by using online tools such as Zoom, Google Meet etc.
- S/he e-mails to the all users of the system by using e-mail addresses in bulk.
- Student representative can add new announcement to the system.
- S/he can manage live events & online meetings which s/he created.
- S/he can attend live events & online meetings organized by others.
- S/he can upload videos, presentations, lecture notes, papers, etc.
- S/he can request the information of all the users from the admin.
- S/he can view the comments and Q&A made under each video. S/he can also manage unavailable comments made by users.
- S/he can access the presentations and watch the videos uploaded by the other users.

### ***Graduate Module:***

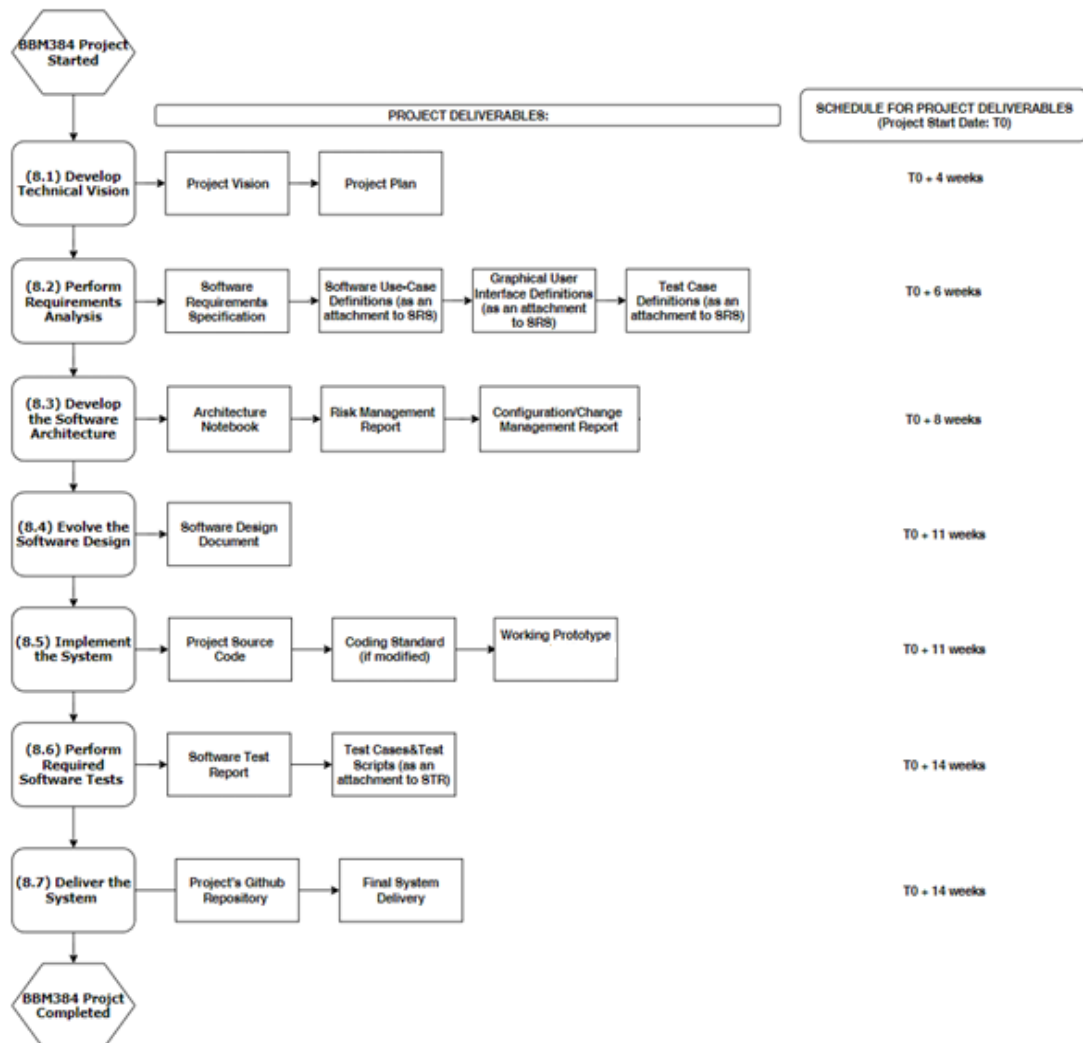
- Graduates can chat with students, academicians and other graduates.
- S/he can enroll to the system and send enrollment request to the admin.
- S/he can make online/offline presentations.
- S/he can write questions and comments under online/offline presentations.
- S/he can add scholarship announcements to the system.
- S/he can add & manage job advertisements to the system.
- S/he e-mails to the all users of the system by using e-mail addresses in bulk.
- Student representative can add new announcements to the system.
- S/he can create live events & online meetings.
  - Live events and online meetings should be scheduled by determining the user's availability via an online tool like when2meet.
  - Online meetings should be done by using online tools such as Zoom, Google Meet etc.
- S/he can manage live events & online meetings which s/he created.
- S/he can attend live events & online meetings organized by others.
- S/he can upload videos, presentations, lecture notes, papers, etc.
- S/he can request the information of all the users from the admin.
- S/he can view the comments and Q&A made under each video. S/he can also manage unavailable comments made by users.
- S/he can access the presentations and watch the videos uploaded by the other users.

**NOTE:** You **MUST add** additional functional requirements different from the described above and **MUST add** non-functional requirements.

### **System Development Constraints**

- The system to develop can be either a web application or a mobile application - not both!
- There must be minimum 9 and maximum 12 use-cases in your system to develop.
- Each group must follow GitHub Flow to keep change and configuration management activities: <https://guides.github.com/introduction/flow/>
  - Each group member must create own branch, get feedback for its own code by creating pull requests, and deploy own code to master branch.
- Model-View-Controller (MVC) architectural pattern must be used for this system.
- JAVA or C# programming language must be used to developed the system.
- For frontend development, groups can use some frameworks such as React, Angular etc.
- Groups can use any database management system such as Oracle, MySQL, SQL Server.
- JAVA or C# programming language must be used to provide connection between the database and the software code.

## 7. Process Flow



## 8. Process Activities

### 8.1. Develop Technical Vision & Project Plan

This activity includes the following tasks.

#### (i) Develop Technical Vision

The students define the vision for the future system by describing the problem and features based on stakeholder requests. The students document the vision as conformant to [Vision Document Template](#).

#### (ii) Develop Project Plan

The students plan the system development by describing the content of the system, project objectives, project organization, development parameters (effort and cost), development

process steps and project milestones. The project plan is documented as conformant to *Project Plan Template*.

## **8.2. Perform Requirements Analysis**

The students analyze the requirements of the system to be developed. The students apply use-case analysis to identify the requirements of the system and document the requirements as conformant to *Software Requirements Specification Template*.

The students document uses cases as conformant to *Use Case Definition Template* or draw activity diagram for each uses case. The students also define graphical user interfaces that the users will interact while executing use cases, as conformant to *Graphical User Interface Design Template*. The students will submit both documents as an attachment to SRS.

At the end of the analysis, the students also define test cases and data for critical use-case scenarios as basis for functional and acceptance tests. The students document test cases as conformant to *Test Case Definition Template* and submit the document as an attachment to SRS.

This activity includes the following tasks.

### **(i) Identify and Outline Requirements**

This task describes how to identify and outline the requirements for the system so that the scope of work can be determined.

The purpose of this task is to identify and capture functional and non-functional requirements for the system. These requirements form the basis of communication and agreement between the stakeholders and the development team on what the system must do to satisfy stakeholder needs. The goal is to understand the requirements at a high-level so that the initial scope of work can be determined.

### **(ii) Detail Use Case Scenarios**

This task describes how to detail use-case scenarios for the system.

The purpose of this task is to describe use-case scenarios in sufficient detail to validate understanding of the requirements, to ensure concurrence with stakeholder expectations, and to permit software development to begin.

### **(iii) Detail System-Wide Requirements**

This task details one or more requirement that does not apply to a specific use case.

The purpose of this task is to describe one or more system-wide requirements (that could not be captured by use-cases and scenarios) in sufficient detail to validate understanding of the requirements, to ensure concurrence with stakeholder expectations, and to permit software development to begin.

#### **(iv) Define Test Cases**

This task requires development of the test cases and test data for the requirements to be tested.

The purpose of this task is to achieve a shared understanding of the specific conditions that the solution must meet.

### **8.3. Develop the Architecture**

The students design, implement, test, and integrate the solution for a number of critical use-case scenarios, and provide a demo of this solution.

The purpose of this activity is to propose a solution to meet the requirements of the system to be developed, and to prove a working prototype of the solution.

The students document architectural details as conformant to [Architecture Notebook Template](#), and provide a demo of the working prototype of their solution.

This activity includes the following tasks.

#### **(i) Design the Solution**

The students identify the elements and devise the interactions, behavior, relations, and data necessary to realize some functionality. The students render the design visually to aid in solving the problem and communicating the solution.

The purpose of this task is to describe the elements of the system so that they support the required behavior, are of high quality, and fit within the architecture.

#### **(ii) Implement Developer Tests**

The students implement one or more tests that enable the validation of the individual implementation elements through execution.

The purpose of this task is to prepare to validate an implementation element (e.g. an operation, a class, a stored procedure) through unit testing. The result is one or more new developer tests.

Developer testing is different from other forms of testing in that it is based on the expected behavior of code units rather than being directly based on the system requirements.

#### **(iii) Implement Solution**

The students implement source code to provide new functionality or fix defects.

The purpose of this task is to produce an implementation for part of the solution (such as a use-case scenario, a class, or component), or to fix one or more defects. The result is typically new or modified source code, which is referred to the implementation.



#### **(iv) Run Developer Tests**

The students run tests against the individual implementation elements to verify that their internal structures work as specified.

The purpose of this task is to verify that the implementation works as specified.

#### **(v) Integrate and Create Build**

This task describes how to integrate all changes made by developers into the code base and perform the minimal testing to validate the build.

The purpose of this task is to integrate all changes made by all developers into the code base and perform the minimal testing on the system increment in order to validate the build. The goal is to identify integration issues as soon as possible, so they can be corrected easily by the right person, at the right time.

#### **(vi) Refine the Architecture**

After performing the tasks above and implementing a working prototype of the architecture, the students may refine the architecture to an appropriate level of detail to support development.

### **8.4. Evolve the Design**

After demonstrating a working architecture of the system and refining the architecture as defined in activity 8.3, the students evolve the architecture into a complete design of the system. The students document system design as conformant to *Software Design Description Template*.

### **8.5. Implement the System**

After proving a prototype of system architecture and defining the design, the students implement the system to meet entire set of system requirements including use case scenarios, system-wide requirements, and non-functional requirements. The students use Software Requirements Specification and follow Software Design Description to carry out this activity.

The students may implement the system as a whole or in iterations. In case of iterative development, the project schedule announced for the delivery of working software iterations will be followed.

The students are expected to develop and follow a Coding Standard while implementing the system. Here is an example *Java Coding Standard*. The students may readily utilize this standard or may define a new one according to their preferences.

### **8.6. Test the System**

The students, from a system perspective, test and evaluate the developed requirements. The students develop test scripts for the selected test cases and document test scripts as

conformant to *Test Script Template*. The students run the tests as defined in the test scripts and document test results as conformant to *Software Test Report Template*.

This activity includes the following tasks. If the students develop the system in iterations, they typically apply these tasks for each iteration.

#### **(i) Implement Tests**

The students implement Test Scripts to validate a Build of the solution. The students organize Test Scripts into suites, and collaborate to ensure appropriate depth and breadth of test feedback.

The purpose of this task is to implement step-by-step Test Scripts that demonstrate the solution satisfies the requirements. The students use Test Case Definition created by activity 8.2 (and delivered as an attachment to SRS) as input to this task.

#### **(ii) Run Tests**

The students run the appropriate tests scripts, analyze results, articulate issues, and document test results.

The purpose of this task is to provide feedback about how well a build satisfies the requirements.

### **8.7. Deliver the System**

The students deliver a complete, defect-free system that meets all the requirements specified in Software Requirement Specification. The students provide source code and relevant data for the system in execution.

The acceptance of the final system is carried out by Teaching Assistants under the supervision of the Instructor. The assistants may run different tests, in addition to the ones specified in the Software Test Report, on the system during acceptance. It is recommended that students ensure defect-free execution of their systems prior to delivery.

## **9. References**

- (i) IEEE std 830-1998 Recommended Practice for Software Requirements Specifications
- (ii) IEEE std 1016-1998 Recommended Practice for Software Design Descriptions
- (iii) IEEE std 829-1998 Standard for Software Test Documentation

## INITIAL USE CASE DIAGRAM

