

**CS251 – Software Engineering I**

|  |  |
| --- | --- |
| **Staff Memebers** | **Email** |
| **Dr Mohamed El Ramly** | **m.elramly@fci-cu.edu.eg** |
| Eng Catherine Bedrossian | c.bedrossian@fci-cu.edu.eg |
| Eng Sara Tarek Ali Othman | s.tarek@fci-cu.edu.eg |
| Eng Mohamed Samir | m.samir@fci-cu.edu.eg |
| Eng Marwa Nabil | m.nabil@fci-cu.edu.eg |
| Eng Yomna Magdy Mohamed | yomna@fci-cu.edu.eg |
| Eng Desoky Abd El-qawy | d.abdelqawy@fci-cu.edu.eg |
| Eng Omar Khaled Ali Ragab | o.khaled@fci-cu.edu.eg |
| Eng Mohamed Elarnaoty | m.elarnaoty@fci-cu.edu.eg |
| Eng Mostafa Saad | m.saad@fci-cu.edu.eg |
| Eng Manar | m.elkady@fci-cu.edu.eg |

**ومن يتق الله يجعل له مخرجًا \* ويرزقه من حيث لا يحتسب ، ومن يتوكل على الله فهو حسبه**

**احرص على ما ينفعك واستعن بالله ولا تَعْجَز ، و لا تقل مستحيل فإن الله على كل شئ قدير**

**Introduction**

* This document states the different project phases and their details.
* Project is 3 phases: Requirements, Design, and implementation and testing.

**Project Logistics**

1. Students from the same lab will be divided into groups; each group is 3~4 members.
2. Project weight is 15 marks from the total course mark.
3. TAs are your clients العميل الذى يدفع مرتبك بتمامه and they will assign you a project. Each TA will have the same project cross all his teams in all labs.
4. See the table below to know TAs Projects
5. You are allowed to choose your project ONLY if you have a real external client and you provide the evidences for that. Then TA will discuss the details with you and must approve the project.
6. Your team will register their names with the TA and CANNOT you change teams after registration.
7. Academic honesty is assumed. All work submitted must be original and written by your team (Not copied from students, the net, outside sources). Plagiarism will be penalized.

* Soon, you will be our colleague and we will be proud of you.
* Professional conduct and practice is essential in your career.

|  |  |
| --- | --- |
| **TA Name** | **Project** |
| Mohamed El Arnaoty | Multiplayer Distributed Game |
| Manar El Kady | Automated Garbage System |
| Desoky | Web-Service to Code |
| Marwa Nabil | FCI Research |
| Mostafa Saad | Learning Management System |
| Yomna Magdy | Books Network |
| Omar Khaled | Chart Component |
| Catherine bedrossian | Products Exchange |
| Mohamed Samir | Social Network |
| Sara Tarek Ali | Event Organizer |

**Project Phases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Deliverables** | **Deadline** | **Mark** |
| Phase 1-a | Initial SRS Document (Function Requirements + Scope only). Furthe | 1 November | 1 |
| Phase 1-b | Final SRS Document | 8 November | 4 |
| Phase 2-a | Initial SDD Document | 15 November | 1 |
| Phase 2-b | Final SDD Document | 2 December | 4 |
| Phase 3-a | Initial Implementation | 13 December | 1 |
| Phase 3-b | Final Implementation | 20 December | 5 |
| Phase 3-c | Project Presentation | 21-25 December | 2 |

# Phase 1: SRS Document

* Each team will be given a document that describes the project problem statement in.
  + TA will act as your product owner.
  + Your role is to understand the main features and requirements of the product.
  + Think about the missing details and discuss them with TA.
  + Ensure that you fully understand what the product owner needs.
  + Do not add any extra major features / components on your own. It is beyond the scope!

# Details

* Each team will read and understand the given project problem statement.
* List **all the functionality** in the project with a number from 1-5 that express the complexity of the functionality. 1 is easy and 5 is complex.
* Make sure to think in any **missing details**, further sub-features and discuss with the TA if needed.
* For each functionality, A Use Case Table will be provided that describes the functionality in details. Document should contain all the functionality covering the requirements.
* You should determine the related non functional requirements and explain them. We expect at least **2 x team size** non functional requirements.
* Use case diagram and any further points in the template should also be filled.
* You will **submit** work on **2 stages**.
  + **In first time**, you will submit documet with **ONLY scope and functioal requirements**. Each TA will send feedback to students from the TA (either direct for your team, or listing common mistakes in the lab).
  + Then, in stage 2, you will submit the updated document considering all the mistakes. Don’t hesitate to ask TA / Visit him in office hours.
* Check the **SRS Document** template attached (CS251-SE2014-Phase 1-SRS-Helpful Example).
  + There is another folder of requirements, they are **optional** for checking.
* Any documents should be uploaded to github 1 day maximum after submission date.

# Phase 2: SDD Document

* In this phase we will work on the Software Design Specification Document.
* SDD document is 5 elements. The 2 major ones are **Class** Diagram and **Sequence** Daigram. Note:
  + **ERD** is translation of class diagram. If your system typically won’t have database, then you shouldn’t provide ERD.
  + **System decomposition** is based on class diagram.
* There will be 2 stages of delivery.
  + In draft submission, teams should provide **all** inital diagrams.
    - However, In Sequence diagrams, you could draw only **2 major** sequecne diagrams.
      * Major is not signin and out. Major is one that in heart of project
    - Also, provide only **2 major** design of user interfaces. Use whatever tools to do them.
  + In final submission, all diagrams should be provided and should be fixed. Requested sequence diagrams are at least **2 x team size** such that each team member does one **major** sequence diagram.
    - Note: Major sequence diagram is for non simple requirements. E.g. Sign in is not major sequence diagram.
    - **Sequence diagram** must reflect and betranslation of **use cases** - u cannot have use cases in one way and sequence in another way
* Check the **SDS Document** template attached (CS251-SE2014-Phase2-SDS-Template.docx).
* Any documents should be uploaded to **github** 1 day maximum after submission date.

# Phase 3: Implementation & Quality Assurance

* The target of this phase is converting the design to implementation while asserting on quality perspective.
* Implementation could be in any language that satisfy OOP. We highly recomend Java.
* Team Should declare:
  + What is the Coding style to follow. Use one given in Lab1, Or search for another.
  + What is the Documentation standard and rules to follow?
* Each TA will upload the best SDS and your implementation will be based on it. In Grading section below, each TA listed what to be implemented for each project.
  + In initial delivery:
    - Team should convert the whole design to code using the design tool and fix all compile errors (typically tools can’t generate 100% running code).
    - Team should pick the **2 major sequences** and convert them to code. If project need a database, it should be created. You don’t need to have every table in this phase. Just the tables needed for the sequence diagram*. In some cases, a TA will specify a different thing (see grading criteria)*
    - If some features out of the major sequences are needed, you **may** think in code workarounds to bypass this problem. E.g. you don’t do login use case, and hard code username and password in the session.
    - Respect project type..if it needs web, develop web. If need GUI, develop GUI
  + In final delivery,
    - Every requirement the TA stated to be implemented should exist from A-Z.
    - Bouns could also be provided.
* Github
  + Each team memeber should show **multiple operations** for the github account.
  + Be ready to be asked how/why you applied the different commands (see section document for git hub).
  + We expect ONLY one repository, NOT one per team memeber. Team leader create one repository and give share to other members (Google “github add collaborator”).
  + Any faking may result in negative grade. Either you used it or did not. NO one allowed to uploaded with your account.
* In general, implementation must adheere to the design document. However, Sometimes, designs are missing some elements (classes, methods or attributes). In case you decicded to add/change something in the design, please **inform** yor **TA**. Submit the updated design with the implementation. Any implementation that will simply ignore the design will get ZERO.

**Presentation**

* **Upload your presentation with final project by 20 Dec 2014 for TA to review it for you**
* Each team will present their work in 5 to 6 minutes, covering the following topics:
  + Overview of the project objectives, requirements, your design and your implementation
    - Do not repeat what other groups say
  + Demo of the system (live or pre-recorded on video)
  + The challenges and difficulties your team faced
  + Lessons your team learned from this project
  + What you would do better or different next time

**Github**

* Every team member must use bit Github account. Any careless behavior won't be accepted (e.g one team member upload data.
* It will be used for documents and code.
* Github history much show real utilization for the project. Any trial to work away of it and upload files in last moments won't be accepted.

**Grading**

# Phase 1: SRS Document

### Draft Submission [1 Grade]

* Seriuos work will be given **full grade**, regardless of some mistakes.
* Work that just do simplifying for work instead of detailing it => **Half grade**
* Work that doesn’t make sense or over simplify document => **Zero grade**

### Final Submission [4 Grades]

* **0.5** grade [Software Purpose, Scope, Definitions].
* **1.5** grade functional
  + 1 grade for correctly converting problem statement requirements. Students shouldn’t miss any required operation
  + 0.5 grade for elaborating on the requirements, listing the **missing details** for them.
  + -0.5 for going out of scope and introducing unreleated major features.
* 0.5 functional requirements
  + **At least 2 x team size** non functional requirements
* 0.5 use case model
  + -0.25 for incorrect include / extend relationships
* 1 use case tables
  + **At least 2 x team size** tables
  + Tables should be very clear. -0.5 for bad flows.

# Phase 2: SDS Documents

### Draft Submission [1 Grade]

* 0.5 for a really good class diagram.
* 0.5 for the other 4 elements in the document
  + Good work will be given **0.5 grade**, regardless of some mistakes.
  + Work that miss elements, up to **0.25 grade**

### Final Submission [4 Grades]

* 1.5 grade for class diagram
  + -0.25 if important algorithms are not listed
* 0.25 grade for system decompsition
* 0.25 for ERD
* 1.25 Sequence Diagrams
  + Should be totally correct. Methods calls with parameters should correspond to actual elements in class diagram.
* 0.25 Class-Sequence usage table.
  + It must be **totally** correct. Few mistakes should take -0.25.
* 0.5 User interface design
  + Students are allowed to draw it using any means.

# Phase 3: Implementation and Testing

### Draft Submission [1 Grade]

* 0.2 Starting to convert Design to code matching the intended design.
* 0.4 for each of the 2 major sequence diagrams.
* Note
  + Please Check below if your TA specifed exactly what to do in inital submission. If something specified, then ignore the above criteria.

### Final Submission [5 Grades]

* Note, most of the below grades out of 4. Each TA will scale the 4 to 5.
* Big penality for not using github by any team memeber

|  |
| --- |
| **Eng Cathrine**:  First screen with Signup Button and Login Button  1- Sign Up 0.5 **grade**  2- Login 0.25 **grade**  3- Logout 0.25 **grade**  Account Home Screen has 3 buttons    4- My Ads, when click on it, see the ads created by this user, see only add Name, Place and Category and when click on the Ad, See all the details, Add Name, description, Ad Info, Personal Info 1 **grade**  5- Create New Ad Button, when click on it, open form to create new Ad 1 **grade**  6- Search Button, when user click on it, see all Ads, and see filters upward,, Place and Category, each a dropdown list with checkboxes and Filter button when click on it perform filtering on the visible Ads 1 **grade**  **Bonus**   * 1 **grade:** make Saving lists button where user can see his saving lists, create new ones, and in the Ad Deatils form, button Add to my saving list will be Added and drop down list of the user saving lists where user enters on which saving list he want to save * 1 **grade:** Delete and Update in Ads created. |
| **Eng Sara:**  1. Starting to convert Design to code matching the intended design. (0.2 initial)  2. Implement Event and User classes with properties as described in class diagram. (initial 0.3)  3. Implement Profile and Settings Classes, connect them with User class where Users  should create their own profiles with settings and their preferences on sign up (pref. To be implemented as part of notification system which is bonus) (initial 0.2)  4. Sign-Up and Sign-In (guided by the sequence diagram). (initial 0.3)  5. Provide categories for evens and connect it with Event class: (final 0.2)   * Art * Music * Tech * Entertainment * Business * Other.   6. User can edits their profile settings and preferences. (final 0.3)  7. Users can:   * create event (final 1) |
| **Eng Marwa:**   * **Authentication: (1 Grade)**   + Sign up >> 0.5     - Note: User must select their interest from the set of available interests in the system   + Login/ Logout >> 0.5 * **Research Component: (2 Grades)**   + Share/ Edit/ Delete Research >> 0.5     - Note: User must specify tags for his research while adding new one   + System send notification for users with specified tags >> 0.5     - Note:       * Notifications should be permanent in the system so user can refer to it later       * User can see the list of received notifications   + User can send "Request Access" to access research, and this yield a new notification for research author >> 0.5   + Research author can reply the notification with (will send mail / Reject) >> 0.5 * **Group Component: (1 Grade)**   + Create / Edit / Delete research group >> 0.5   + View available groups >> 0.5 * **Bonus: (2 Grades)**    + Handling all notifications by mail: (1 Grade for any of them)     - System send mail for newly added researches to interested users     - System send mail for "request access"     - User can receive the author reply by email   + Upload research (1 Grade)   + Search component in the system (1 Grade)     - User can search in researches, users, and groups     - User can specify search criteria: tags, author name, ..   + Professional user interface (1 Grade)   + Note: Pick up two requirements of the four above |
| **Eng Mostafa:**   * Project is a website (e.g. may use java JSP/Servlets...or strong framework such as JSF). * Database should exist with the relative tables of below features (not every table). * All points below are simple web prgoramming of add/update in database quires. Nothing complex. For **upload**, google web available components to do that for you. * Register / Sign in For Student and Doctor (not TA) => **0.5 grade.** * Create Course => 0.25 grade. List Courses **0.25 grade**. * Student Registeration in a course: **0.25 grade.** * Board simple posting only (**0.25 grade**). By simple, I mean no one replies on you. Alowing replies on owner post **0.5 grade**. This is not complex as acadox, you reply in serial way such as forums/gmail/hotmail, NOT reply on specific reply, which was another reply....and so on. * Create normal assignment and uploading its description file. **0.75 grade**. * Create programming assignment and uploading its description file and set of input and output files. **0.5 grade**. * Student uploads answer of an assignment (**0.25 grade**). * Dr list students submissions of an assignment (**0.25 grade**). Select one of them to download (**0.25 grade**). Set grade for a student assignment (**0.25 grade**)   **Bonus**:   * Compile and Judge programming assignment and setting its grade (assuming c++ or java) **1 grade** * When someone replies on a board post, the own gets an email in his inbox (e.g. his gmail). **0.5 grade** |
| **Eng Youmna:**  Per member: 1) **1.25** contribution in the basic functions (sign-in,signup, view books,search,add book) 2) **2.75** for individually implement two functions of the system 3) if bonus, per member +1 for implementing additional function(i.e. >2 functions)   \*Functions:   1. Mark book as read/to read/currently reading 2. Add book to custom shelves 3. Update progress 4. Rate book/ Write review of a book 5. Like a book review / Comment in a book review 6. Add Friend 7. View his/her profile, Edit basic profile info 8. View book ( basic info, all reviews (sorted by) , average ratings) 9. View news feeds (filter options) : friends updates (sorted by), discussions updates 10. Add quote(additional) (bonus implementation) 11. Become fan of an author (author will be added to user favorites) 12. Compete with one friend for read a book     1. update progress     2. comment in progress     3. view competition result ( who finish earlier is the winner) 13. Create group     1. add members (if admin)     2. add posts (all members)     3. comment (all members) |
| **Eng Mohamed Samir:**   * SignUp, Login **(0.3 grade)** * Ensure that only one user is active **(0.3 grade)** * Users functions (sendFriendRequest(), addFriend(), … ) **(0.3 grade)** * Group Functions **(0.1 grade)** * Group privacy operations ( getAllowedUsers() ) **(0.4 grade)** * Assign privacy to each group **(0.4 grade)** * Post operations **(0.2 grade)** * Post privacy operations **(0.4 grade)** * Handle share post operation (one post may has 2 privacy) **(0.4 grade)** * Page operations (number of likes of each page) **(0.3 grade)** * Message operations (one to one message, one to group message) **(0.3 grade)** * Hashtag operations (count, search, sort) **(0.3 grade)** * DB subsystem, use arrays or vectors to save data **(0.3 grade)**   **Bonus:**   * Make SQL db (or any other suitable db) according to your ERD and connect this db to your API, to use it in your db subsystem, Save data of all 6 components **(1 Bonus grade)**   Handle timeline operations and show requested timeline posts and activities to current login user **(1 Bonus grade)** |
| **Eng Omar Khaled:**   Select Chart  **(0.25 grade)**   Fill the Chart Data **(0.25 grade)**   Check the Chart fill with right data  **(0.5 grade)**   Generate axis **(0.4 grade)**   Generate legend  **(0.4 grade)**   Set chart-features( chart title, legend-position, legend-background ,axis labels) **(0.3 grade)**   Draw legend **(0.4 grade)**   Draw pie-Chart **(0.5 grade)**   Draw bubble-chart **(1 grade)**  **Bonus:**   * Draw line-chart  **(1 grade)** |
| **Eng Mohamed Arnaoty:**  Grading:  - Not conforming to SRS document is penalized by -2  - Not conforming to SRS document is penalized by -0.5 to -4 according to the level of variation  - Game must be multi-player distributed, other wise you will be penalized by -2  - Game that is not running will be penalized by -1.5.  - Game that is running with poor reliability and performance will be penalized by -0.5  - Game Main Screen, and new game screen (0.5 marks)  - Development of Communication protocol:           - Document describing communication protocol and client server messaging format (0.25 marks)           - Protocol coding (0.75 marks)          - Game client and server classes(0.5 marks)  - Game Logic:           - Game Controller classes(1 marks)           - Game Canvas (0.5 marks)  - Team Co-operation   (0.5 marks)  - Code Quality  (0.5 marks)  Bonus:  -0.5 bonus for java doc.[May be change...please see updated versions]  Note:  Whole grade will be scaled to 5 |
| **Eng Manar Elkady:**   * Login, logout (for both customers and manager) **(0.2 grade)** * Customer registration **(0.5 grade)** * The ability to reserve parking **(0.5 grade)** * Users will have ability to cancel or modify reservations **(0.5 grade)**  |  | | --- | | * Read License plate number/ car entrance / car exit by the License Plate reader. **(0.7 grade)** * Interaction with arrival customer: Display the assigned spot / Enter reservation number via keypad. **(0.5 grade)** * View parking garage statics such as reports, no. of spots occupied and no. of spots empty. **(0.5 grade)** * Handling the overstay operation in case of the parking lot is full or is not full. **(0.5 grade)** * Manage Prices by the manager. **(0.4 grade)** * Build the software as a web application. **(0.5 grade)** * Connect the software with a DB. **(0.2 grade)** |   **Bonus:**  **Build a mobile application version to used by the valet to verify reservations and move the cars (1 Bonus grade)** |
| **Eng Desoki:**  1- main form GUI [1]  2- Java Code Generation logic [3] 3- give an option to generate java documents [1 grade bouns]  Note: May be soon more fine details is here... please see updated versions |

### Project Presentation [2 Grade = 20 points]

* Quality of the presentation content and organization **8 marks**
  + Content is clear and at suitable level of details
  + Flow of ideas was smooth and clear from intro to content to conclusion
* Quaity of presentation style **6 marks**
  + The presenters exhibited confident enthusiasm about the subject material
  + The presenters spoke professional, clearly and with suitable speed
  + The presenter interacted effectively and made good visual contact with the audience
* Visual Aids **6 marks**
  + Slides were easy to read and designed with an interesting format
  + Slides were relevant and communicated content effectively

**Policy Regarding Plagiarism:**

**Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.**

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
5. إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر.

**إذا خاطبتك نفسك وقالت**

**لاأعرف فقولي لها يانفس تعلمي**

**وإن قالت لا أقدر فقولي لها حـــاولي**

**وإن قالت مســـــــــتحيل فقولي لها جربي**

**وإن قالت جربت فقولي اســــــتمري وواصلي**

**وإن قالت لم ينفع فقولي غيري واســـــــــــــــتبدلي**

**وإن قالت أخاف أن ينتقدني الناس فقولي لها وما الجديد**

**فقد سُب خير البشر موسى و عيسى و محمد عليهم السلام**