

**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 |

**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 | 4 |

# 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.
    - 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 is converting the design to implementation while asserting on quality perspective.
* Team Should declare:
  + What is the Coding style to follow?
  + What is the Documentation standard and rules to follow?
* Any implementation that doesn't adhere to the design document will be given **zero**. It is not matter of code that do the task, but code that follows the design. For the design patterns, you should implement at least 2 of the design patterns.
* Bad Code is useless. Great attention for quality assurance should be given.

**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: SDD 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

**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**