Must have cover page with CU logo, FCI logo, course name, prog and TA, 2014, course name, proj name, leader name, SRS Doc title, contact of leader – Title page has no header or footer- docs must be supplied as pdf – files re-named as requested

Contents

[Instructions [To be removed] 1](#_Toc401308926)

[Team 1](#_Toc401308927)

[Document Purpose and Audience 2](#_Toc401308928)

[Introduction 2](#_Toc401308929)

[Software Purpose 2](#_Toc401308930)

[Software Scope 2](#_Toc401308931)

[Definitions, acronyms, and abbreviations 2](#_Toc401308932)

[Requirements 2](#_Toc401308933)

[Functional Requirements 3](#_Toc401308934)

[Non Functional Requirements 3](#_Toc401308935)

[System Models 3](#_Toc401308936)

[Use Case Model 3](#_Toc401308937)

[Use Case Tables 3](#_Toc401308938)

[Ownership Report 4](#_Toc401308939)

[**Policy Regarding Plagiarism:** 5](#_Toc401308940)

# Instructions [To be removed]

* **IMPORTANT. Rename this document to CS251-TAName-LeaderID-SRSDocument.docx**

**(e.g. CS251-MostafaSaad-20040752-SRSDocument.docx)**

* **Remove the following notes and any red notes**
* **This document is the template document for your SRS.**
* **Read the accompanied example(s). Read Chapter 4 from OO SE, 2nd ed., by Tim Lethbridge**

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
|  | 1st name is team leader |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Purpose and Audience

* **Any document anywyere should tell us 2 things: What is this document? Who is execpetd to read it?**
* **Write in simple notes, What is this documet?**
* **List the taget audience to read this document (e.g. CEO? Project Manager? Customer...?)**

# Introduction

## Software Purpose

* **Summarize the purpose of the software**

## Software Scope

* **Any software could have too many componeonts / Major features .. but we should implement specific things...this is the scope**
* **In simple points, what is the software scope (focus on componeonts / Major features, not tiny things)**

## Definitions, acronyms, and abbreviations

* **In a table, list all needed ones. Consider the audeicne**
* **Think as following: Document has abbreviation ATM..IFF audience doesn’t know it, let’s calrify it.**

# Requirements

## Functional Requirements

* **This is the most critical part...** **functional requirements describe what the system should do**
  + **E.g. an ATM allows you to enter Card, enter user name password and withraw a money**
* **List all the system requirements, respecting the problem statement giving by your TA**
  + **Make sure to go in the missing details for the mentioned features/components**
    - **Discuess with TA**
  + **Going byeond them (e.g. adding new complete major feature / component) is breaking the statement scope**
* **Each requirement should be clearly described, such that it can be understood without the presence of the one who wrote it.**

## Non Functional Requirements

* **Non-functional requirements describe how the system works**
  + **E.g. Withdraw operation will be done within 20 second. Network is using secured protocols. System allows up to 30,000 withdrawal per minute.**
  + **Think about the operation / system quality**
* **There are too many non functional requirements. Read in** [**wiki**](http://en.wikipedia.org/wiki/Non-functional_requirement) **and Chapter 4. Pick the suitable ones for your system. Non-functional requirements must be VERIFIABLE, i.e., MEASURABLE.**
  + **Some Types as just examples: Usability, Reliability, Performance, Security, Scalability, Portability, Maintainability**
  + **Select the sutiable ones, for each one write the details**
  + **Be realistic ☺**

|  |  |
| --- | --- |
|  | **Details** |
| **Performance** | * **Withdraw operation will be done within 20 second** |
| **Scalability** | * **System could be horizontally ....** * **And** * **In addition** |

# System Models

## Use Case Model

* **Using UML, write the use case model expressing the system actors & operations**

## Use Case Tables

* **Using below table template, for each requirement write a use case table that shows user/system interaction**
  + **If one requirement is so big, you could divide it to more than table**
  + **If some requirements are not major, you could plugin them in other senario**
    - **E.g. you may not do LogIn Usecase table as it is simple functionality**
* **Flow of events should be very detailed**

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: |  | |
| Actosr: |  | |
| Pre-conditions: |  | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enter Card and Password. |  |
|  | 2- System Verify user data |
| 3- User Select Vodafone from the list |  |
|  | 4- System retrieves Vodafone bills |
| and so on |  |
| Exceptions: |  | |
| Includes: |  | |
| Notes and Issues: |  | |

# Ownership Report

* **Remove the following notes and any red notes**
* **For every item in this document, write the owners. If someone is owner of something, s/he understands it 100.%**
* **Team leader must verify the table with the team members.**

|  |  |
| --- | --- |
| **Item** | **Owners** |
|  |  |
|  |  |

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.