

ENSE 374, Software Engineering Management (Fall 2018)

Milestone 1 – Extended requirements digging & empathy mapping

Due: September 27, 2018

Allocation: 4%

Activities: Following the “understand” phase of Design Thinking, with your team extend the original requirements document titled “Project high-level requirements (charter and vision)” that is available in URCourses using the template provided in Appendix A (**NOTE: you will not fill out all the sections/information in the template provided in Appendix A for this Milestone. Focus on Sections 1, 2, 4.2, 4.5, 5.1.1, 5.1.2, 5.4.1 (etc.) – Remember, this is just a template. We/Your team may not use all sections**). In the spirit of Design Thinking methodology, ensure to document your team’s (new) foundational problem statement and any new ideas, new information, and/or new directions that your team is going to take for the project. To help gather user information, create empathy maps based on the discussions you all experienced in the first lab time (on Sept 19/21 respectively). Keep things at a relatively high-level for now, i.e. you will be going through the other phases of Design Thinking and expanding your functional requirements document in future milestones. This means although you can chat about ideas your team may have for the “explore” and “materialize” phases of Design Thinking, there is no need to really flesh out all the details of those phases yet (the emphasis of this milestone is the “understand” phase). Develop slides for **your team’s in-class report-out on September 27** using the supplied template addressing, at a minimum, all key criteria as listed within.

Fail Grade: 0%	Developing Grade: 1%	Meets expectations Grade: 2-3%	Exceeds expectations Grade: 4%
Not submitted (group) Non-attendance during report-out (individual)*	Extended requirements digging, empathy map(s), and other activities are weak/missing key elements Unreadable content (Requirements, problem definition, empathy maps, or slides) All files are uploaded to GitHub Report-out exceeds 10 minutes**	Good extension and representation of extended requirements digging, empathy map(s), and other activities Readable content (Requirements, problem definition, empathy maps, or slides) All files are uploaded to GitHub Report-out is within 10 minutes**	All within “Meets expectations” Meets Tim’s “wow” factor (Above and beyond)

NOTES: Attendance during group report-outs is mandatory. Failure to attend will result in a 0% grade on the milestone for the individual that is missing. Group members in attendance during the report-out will be graded accordingly as illustrated above. Time allocated for report-outs are not to exceed 10 minutes. Your group will be given a “1-minute remaining” notify. Please ensure 1:00-2:00 minutes for questions/answers within the allotted 10 minutes. I will randomly select the group presentation order on report-out day. If a group member is not present, I will ask another group to present. The person not present will be deducted 0.5% from their milestone grade (individual deduction, not group). If I ask again and the person is still not present, the group will proceed with the report-out and the person not present will receive a grade of 0% on the milestone (individual deduction, not group). Please also note that a group member will also not be permitted to join their presentation late (i.e. if it’s already begun). The Instructor reserves the right to have a grading discretion +/- 1%



Appendix A: Functional Requirements Document (FRD) Template

Template from: <http://almooc.com/downloads/FRDTemplate.doc> (Accessed September 2018)

TITLE: Faculty of Engineering URSource Functional Requirements Document

Team name:

Team members:

GitHub URL:

Version	Description of Change	Author	Date

1 INTRODUCTION

[Provide an overview of the system and some additional information to place the system in context.]

1.1 Purpose

[Provide an overall description of the FRD, its purpose. Reference the system name and identifying information about the system to be implemented.]

1.2 Scope

[Discuss the scope of the document and how it accomplishes its purpose.]

1.3 Background

[Describe the organization and its overall responsibilities. Describe who is producing the document and why.]

1.4 References

[List references and controlling documents, including: meeting summaries, white papers, other deliverables, etc.]

1.5 Assumptions and Constraints

[Provide a list of contractual or task level assumptions and/or constraints that are preconditions to preparation of the FRD. Assumptions are future situations beyond the control of the project, whose outcomes influence the success of a project.]

1.5.1 Assumptions

Examples of assumptions include: availability of a technical platform, legal changes and policy decisions.



1.5.2 Constraints

Constraints are boundary conditions on how the system must be designed and constructed. Examples include: legal requirements, technical standards, strategic decisions.

- Constraints exist because of real business conditions. For example, a delivery date is a constraint only if there are real business consequences that will happen as a result of not meeting the date. If failing to have the subject application operational by the specified date places the organization in legal default, the date is a constraint.
- Preferences are arbitrary. For example, a date chosen arbitrarily is a preference. Preferences, if included in the FRD, should be noted as such.]

1.6 Document Overview

[Provide a description of the document organization.]

2 METHODOLOGY

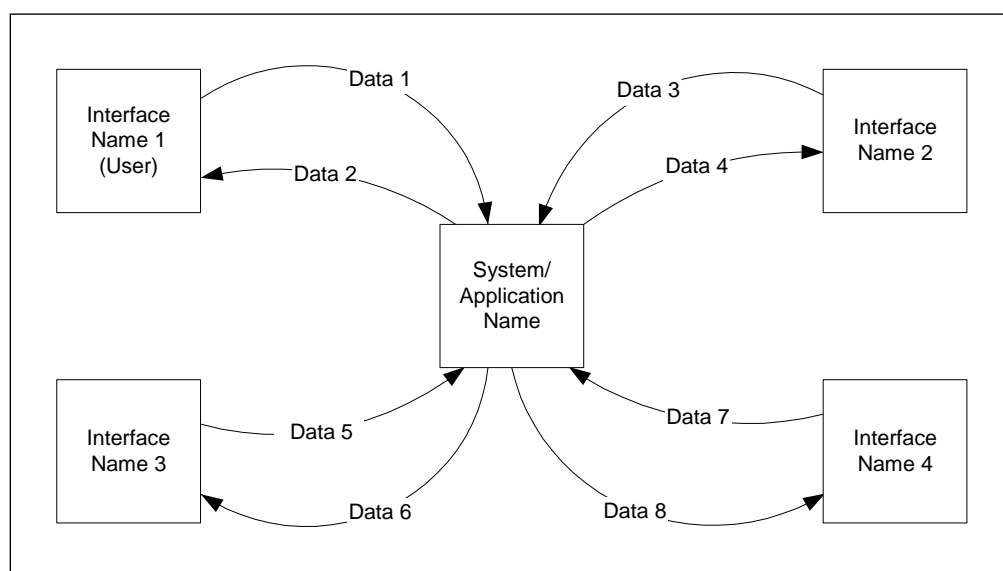
[Describe the overall approach used in the determination of the FRD contents. Describe the modeling method(s) so non-technical readers can understand what they are conveying.]

3 FUNCTIONAL REQUIREMENTS

4.1 Context

[Provide a context diagram of the system, with explanations as applicable. The context of a system refers to the connections and relationships between the system and its environment.]

Exhibit 2 - Generic Context Diagram





4.2 User Requirements

[Provide requirements of the system, user or business, taking into account all major classes/categories of users. Provide the type of security or other distinguishing characteristics of each set of users. List the functional requirements that compose each user requirement. As the functional requirements are decomposed, the highest level functional requirements are traced to the user requirements. Inclusion of lower level functional requirements is not mandatory in the traceability to user requirements if the parent requirements are already traced to them.]

User requirement information can be in text or process flow format for each major user class that shows what inputs will initiate the system functions, system interactions, and what outputs are expected to be generated by the system. The scenarios should be comprehensive, to the extent that all user types and all major functions are covered. Give each user requirement a unique number. Typically, user requirements have a numbering system that is separate from the functional requirements. Requirements may be labeled with a leading “U” or other label indicating user requirements.]

4.3 Data Flow Diagrams

[Decompose the context level diagrams to determine the functional requirements. Data flow diagrams should be decomposed down to the functional primitive level. These diagrams are further decomposed during design.]

4.4 Logical Data Model/Data Dictionary

[Create the initial Logical Data Model. Describe data requirements by providing data entities, decomposition, and definitions in a data dictionary. The data requirements describe the business data needed by the application system. Data requirements do not describe the physical database and are not at the level of identifying field names.]

4.5 Functional Requirements

[List the functional requirements of the system.]

4.5.1 Functional Requirements Group 1

[List the functional requirements for each functional requirements group.]

Exhibit 4 - Sample Requirements Group 1

Section/ Requirement ID	Requirement Definition
FR1.0.	The system shall [parent requirement group 1].
FR1.1	The system shall [child/parent requirement].
FR1.1.1	The system shall [child requirement].
FR1.1.2	The system shall [child requirement].



4.5.2 Functional Requirements Group 2, Etc.

5 OTHER REQUIREMENTS

[Describe the non-behavioral requirements.]

5.1 Interface Requirements

[Describe the user interfaces that are to be implemented by the system.]

5.1.1 Hardware Interfaces

[Define hardware interfaces supported by the system, including logical structure, physical addresses, and expected behavior.]

5.1.2 Software Interfaces

[Name the applications with which the subject application must interface. State the following for each such application: name of application, external owner of application, interface details (only if determined by the other application).]

It is acceptable to reference an interface control document for details of the interface interactions.]

5.1.3 Communications Interfaces

[Describe communications interfaces to other systems or devices, such as local area networks.]

5.2 Data Conversion Requirements

[Describe the requirements needed for conversion of legacy data into the system.]

5.3 Hardware/Software Requirements

[Provide a description of the hardware and software platforms needed to support the system.]

5.4 Operational Requirements

[Provide the operational requirements in this section.]

Do not state how these requirements will be satisfied. For example, in the Reliability section, answer the question, “How reliable must the system be”? Do not state what steps will be taken to provide reliability.

Distinguish preferences from requirements. Requirements are based on business needs, preferences are not. If, for example, the user requires a special response but does not have a business-related reason for it, that requirement is a preference.

Other applicable requirements on system attributes may be added to the list of subsections below.]

Operational requirements describe how the system will run and communicate with operations personnel.

5.4.1 Security and Privacy

[Provide a list of the security requirements using the following criteria:

A. State the consequences of the following breaches of security in the subject application:



1. Loss or corruption of data
 2. Disclosure of secrets or sensitive information
 3. Disclosure of privileged/privacy information about individuals
 4. Corruption of software or introduction of malware, such as viruses
- B. State the type(s) of security required. Include the need for the following as appropriate:
1. Physical security.
 2. Access by user role or types.
 3. State access control requirements by data attribute. For example, one group of users has permission to view an attribute but not update it while another group of users has permissions to update or view it.
 4. State access requirements based on system function. For example, if there is a need to grant access to certain system functions to one group of users, but not to another. For example, "The system shall make Function X available to the System Administrator only".
 5. State if there is a need for certification and accreditation of the security measures adopted for this application]

The Security Section describes the need to control access to the data. This includes controlling who may view and alter application data.

5.4.2 Audit Trail

[List the activities recorded in the application's audit trail. For each activity, list the data recorded.]

5.4.3 Reliability

- A. [State the following in this section:
1. State the damage can result from failure of this system—indicate the criticality of the software, such as:
 - a) Loss of human life
 - b) Complete or partial loss of the ability to perform a mission-critical function
 - c) Loss of revenue
 - d) Loss of employee productivity
 2. What is the minimum acceptable level of reliability?
- B. State required reliability:
1. Mean-Time-Between-Failure is the number of time units the system is operable before the first failure occurs.
 2. Mean-Time-To-Failure is the number of time units before the system is operable divided by the number of failures during the time period.
 3. Mean-Time-To-Repair is the number of time units required to perform system repair divided by the number of repairs during the time period.]



Reliability is the probability that the system processes work correctly and completely without being aborted.

5.4.4 Recoverability

[Answer the following questions in this section:

- A. In the event the application is unavailable to users (down) because of a system failure, how soon after the failure is detected must function be restored?
- B. In the event the database is corrupted, to what level of currency must it be restored? For example “The database must be capable of being restored to its condition of no more than 1 hour before the corruption occurred”.
- C. If the processing site (hardware, data, and onsite backup) is destroyed, how soon must the application be able to be restored?]

Recoverability is the ability to restore function and data in the event of a failure.

5.4.5 System Availability

[State the period during which the application must be available to users. For example, “*The application must be available to users Monday through Friday between the hours of 6:30 a.m. and 5:30 p.m. EST.* If the application must be available to users in more than one time zone, state the earliest start time and the latest stop time. Consider daylight savings time, too.

Include use peak times. These are times when system unavailability is least acceptable.]

System availability is the time when the application must be available for use. Required system availability is used in determining when maintenance may be performed.

5.4.6 General Performance

[Describe the requirements for the following:

- A. Response time for queries and updates
- B. Throughput
- C. Expected rate of user activity (for example, number of transactions per hour, day, or month, or cyclical periods)

Specific performance requirements, related to a specific functional requirement, should be listed with that functional requirement.

5.4.7 Capacity

[List the required capacities and expected volumes of data in business terms. Do not state capacities in terms of system memory requirements or disk space—if growth trends or projections are available, provide them]

5.4.8 Data Retention

[Describe the length of time various forms of data must be retained and the requirements for its destruction.

For example, “The system shall retain application information for 3 years”. Different forms of data include: system documentation, audit records, database records, access records.]



5.4.9 Error Handling

[Describe system error handling.]

5.4.10 Validation Rules

[Describe System Validation Rules.]

5.4.11 Conventions/Standards

[Describe system conventions and standards followed.

For example: Microsoft standards are followed for windows, Institute of Electrical and Electronics Engineers (IEEE) for data formats, etc.]

APPENDIX A - GLOSSARY

[Define terms, acronyms, and abbreviations used in the FRD.]