

## LAB 7

---

### Objectives

1. Identify the entities of a detailed design for your project.
2. Create detailed designs for a subset of the entities of your project.
3. Assess your team's technical capability compared to the technical needs of the project.

Designers need to specify the details of the entities that make up the system. These definitions should be sufficiently detailed that the design can be given to a developer and the developer can create the entity as envisioned by the designer.

Once your team starts to develop a design, you should also be developing a better understanding of the technologies and skill levels needed to build the product. As a separate task, this lab will also provide a chance for you to assess your team's capability to work on the project and identify learning or skill development you may need.

### Procedure

#### Step 1 – Draft a list of entities for your project

You should consider the following types of entities:

- Screens (or Web pages)
- Database tables
- Files (e.g., data that is stored as part of the system but not stored in a database)
- Code (modules, objects, or functions)

Use Figure 7-1 to list all the system entities that you can identify. A good way to start is to pick one area and focus on that. For example, if your system has a significant user interface, start by trying to name all the screens that would comprise your interface. For each entity you list:

- Enter a type, e.g., “screen”
- Give it a meaningful name, e.g., “CustomerProfile”
- Provide any short notes or explanation needed to identify the screen, e.g., “This screen captures customer information and preferences.”

#### Step 2 – Create detailed designs for at least 4 of your entities.

You will not be able to design all the entities of your system in this lab, but this step will get you started. Pick 4 entities that you think you understand the best at this point, and create a design for them. Every entity should have a name, type, and design details. Templates are provided to help you create detailed design for screens, database tables, and code functions.

#### Step 3 – Review your detailed designs.

After creating your designs, review them for completeness and clarity. Ask yourself this question: “If I was the developer and a designer handed me this design, would I know what to build without needing to ask a lot of questions?”

If you have created the design entities as a team, set them aside for a few minutes before review each one. If you have worked in sub-groups within your team to create the designs, then exchange designs so the reviewer is a different person than the creator of a design.

Revise your designs based on the review.

**Step 3 – Assess your team’s capability to complete this project.**

Once you have an architectural overview and the beginning of a design, you should be able to assess capability and identify things that someone on the team may need to learn. Use Figure 7-5 to summarize this information.

3.A – List the technologies you need for your project using the column on the left. Consider things such as programming languages, operating systems, specialized data sources, software libraries, support tools, and hardware.

3.B – List each team member at the top of a column, and then evaluate that person’s knowledge of the technology in each row. For the column for each team member, use the following values:

- 1 – No knowledge or not much relative to the needs of this project
- 2 – Enough knowledge to accomplish part but not all of this project
- 3 – Knowledge probably sufficient for this project

3.C – Discuss within your team how you will start to gain capabilities that you are missing. You do not need to turn in results of this discussion in this lab, but will need to address this in the coming weeks.

**What to Turn In**

In order to obtain full credit for this lab, **each team** must turn in:

1. Figure 7-1 – Possible System Entities
2. Detailed designs for at least 4 entities in your system. Use the templates in Figures 7-2 through 7-4 to get started.
3. Figure 7-5 – Team Capability Assessment

**Figure 7-1 – Possible System Entities**

Product: &lt;Enumeration Mathematical Library&gt;

Team: &lt;707&gt;

Date: &lt;02/23/2016&gt;

Type	Name	Description or Notes
Screen	Website	Our website enumeration.ml for hosting our software
Database	Graphic Base	Our database of mathematical basic graphics
Database	Number Base	Database of important mathematical numbers and formulas
File	Makefile	For compiling and building software
File	num.c	Common number theoretic functions are in this file
File	alg.c	Algebraic number theory structures and functions are located in this file
File	combi.c	Combinatorial structures and functions are located in this file
File	field.c	Field types and operations are located in this file
File	arb.c	Arbitrary precision numbers type and operations are located in this file
File	graphic.c	For Graphics Renderings
File	num.h	Header file for num.c
File	alg.h	Header file for alg.c
File	combi.h	Header file for combi.c
File	field.h	Header file for field.c
File	arb.h	Header file for arb.c
File	graphic.h	Header file for graphic.c
File	math.s	Fast math functions in assembly language
File	operation.s	Fast math operations in assembly language
File	Function.s	Optimized functions in assembly language
File	eml.tar.gz	Standard Archive distribution of software
File	eml.so	Dynamically Linked Library Version
File	eml.a	Statically Linked Library Version
Module	C++ Bindings	Bindings for C++ Programming Language
Module	Python Bindings	Bindings for Python Programming Language
Module	Haskell Bindings	Bindings for Haskell Programming Language
Module	Java Bindings	Bindings for Java Programming Language
Code	Tex Renderer	Tex Renderings
Code	PNG Renderer	PNG Renderings

## Figure 7-2 - Template for Detailed Design for a Screen

CI102-Lab7(1)

**Name:** <Website (enumeration.ml)>

**Type:** Screen

**Purpose:** This screen is needed to meet requirement 1 of Software Distribution

**Description:** Figure <1> shows the layout for this screen. This screen is the first webpage of our website which we intend to use for our software distribution and propagation.

The screen contains the webpage hierarchy.

Layout:

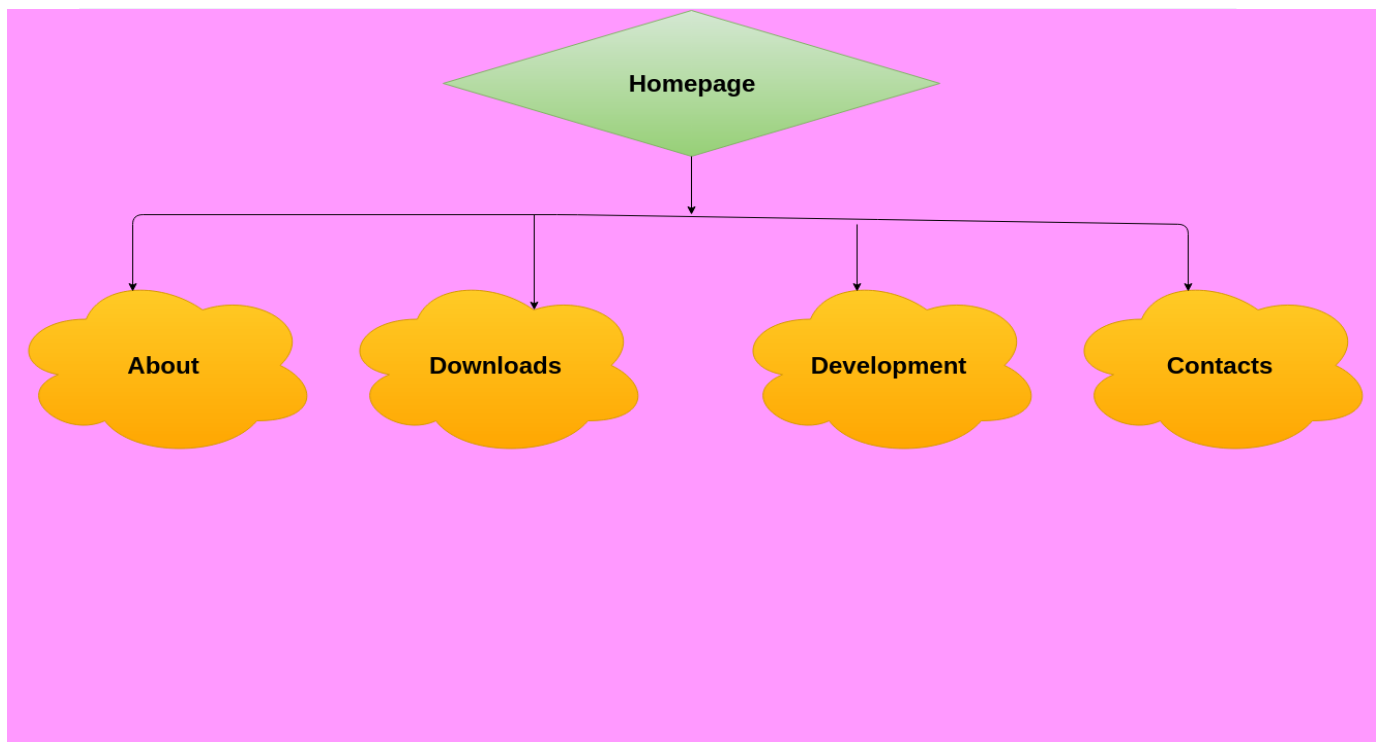


Figure <1> - <Website> Screen

### Figure 7-3 - Template for Detailed Design for a Database Table

**Name:** <Number Base>

**Type:** Database Table

**Purpose:** This table is needed to meet requirement 10.

**Description:** Figure <2> shows the contents for this table. This table contains information about what our database will typically store. One row of this table represents a number important in some mathematical sense.

**Table Contents:**

Data Element Name	Data Type	Key	Notes
Pi	arb_double	1	This is number Pi computed upto any precision.
e	arb_double	2	This is number e computed upto any precision.
Euler's Constant	arb_double	3	This is number Euler's constant computed upto any precision.
The Golden ratio	arb_double	4	This is number golden ratio computed upto any precision.
Glaisher–Kinkelin constant A	arb_double	5	This is number Glaisher–Kinkelin constant A computed upto any precision.
i	arb_complex	6	Complex unit upto arbitrary precision
Sqrt (x)	arb_func	7	Arbitrary precision functions
Complex Integrals	arb_integral	8	Arbitrary precision integrals
Prime Sieve	arb_array	9	Array of prime numbers
Field1	arb_field	10	Hilbert fields
pFields	arb_field	11	P-adic Fields

**Figure <2> - <Number Base> Database Table**

**Figure 7-4 - Template for Detailed Design for a Code Function**

**Name:** <Tex Renderer>

**Type:** Function

**Purpose:** This function is needed to meet requirement 17.

**Parameters:** The following parameters are used to call this function:

Name	Data Type	Notes
data	Any arb datatype	Structure to be Tex rendered
width	double	Relative width to standard width
height	double	Relative width to standard height

**Return Type:** <int>

**Processing:** Takes a arb datatype data structure and output a Tex file to represent it. Variables width and height are taken into account while generating the Tex output. This program takes call tex-generator open source library.

**Figure 7-5 – Team Capability Assessment**

	<Debdut Karmakar>	<Mosfiqur Rahman>
<b>Management</b>	1	3
<b>C Programming</b>	3	3
<b>Assembly Programming</b>	2	2
<b>Bindings</b>	2	1
<b>Math Algoritms</b>	3	2
<b>Graphics &amp; Tex</b>	2	2

\*\* The table values represent an assessment of team member capabilities. The values are:

- 1 – No knowledge or not much relative to the needs of this project
- 2 – Enough knowledge to accomplish part but not all of this project
- 3 – Knowledge probably sufficient for this project