# Project Phase I: KWIC Architecture Specification

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author(s) |
| 9/29/2015 | 1.0 | Preliminary version of K.W.I.C system architecture | All |
| 10/15/2015 | 2.0 | Updated version with professor suggestion from interim presentation | All |

Table of Contents

[1. Introduction 5](#_Toc433093662)

[1.1 Purpose 5](#_Toc433093663)

[1.2 Scope 5](#_Toc433093664)

[1.3 Definitions, Acronyms, and Abbreviations 6](#_Toc433093665)

[1.4 Project Deliverables 6](#_Toc433093666)

[2. PROJECT ORGANIZATION 7](#_Toc433093667)

[2.1 Process Model 7](#_Toc433093668)

[2.2 Organizational Structure 7](#_Toc433093669)

[2.3 Work Schedule 7](#_Toc433093670)

[3. Software Architecture 8](#_Toc433093671)

[3.1 Abstract Data Type Architectural Style 8](#_Toc433093672)

[4. Traceability Matrix 9](#_Toc433093673)

[Functional Requirement and Software Architecture Traceability Matrix 9](#_Toc433093674)

[5. Prototype/Screen Shots 10](#_Toc433093675)

[6. References 10](#_Toc433093676)

# Introduction

The Project that our team will be working on is a web search engine.

A web search engine is a web-based tool that is designed to search and locate information on the World Wide Web. Popular examples includes Google, Yahoo! and Bing. These Search engines utilize automated software applications (referred to as robots, bots, crawlers or spiders) that travel along the Web, following links from page to page, site to site. The information gathered by the spiders is used to create a searchable index of the Web. The search results are generally presented in a line of results often referred to as search engine results pages (SERPs).

## Purpose

The purpose of our project is to develop KWIC index System (Keyword In Context) proposed by David Parnas in early 70’s using Java Applet. This System provides a convenient search mechanism for information in a long list of lines, such as book titles, or online documentation entries.

Parnas described the KWIC problem as follows:

*“The KWIC index system accepts an ordered set of lines; each line is an ordered set of words, and each word is an ordered set of characters. Any line may be “circularly shifted” by repeatedly removing the first word and appending it at the end of the line. The KWIC index system outputs a list of all circular shifts of all lines in alphabetical order.”*

In his paper of 1972, Parnas used the problem to contrast different criteria for decomposing a system into modules. Our team followed the same phenomena to implement KWIC system, by designing the system with 5 highly cohesive modules (input, line storage, circular shift, alphabetical sort, and output). We analyzed functional and nonfunctional requirement, design architecture styles, implement using Java applet and test the system. The KWIC system architecture style shall be an Abstract Data Type (ADT) style as this will provide clear object oriented structure with desire qualities of high cohesion and low coupling.

## Scope

The KWIC system shall be designed, implemented and tested to satisfy list of functional and nonfunctional requirements. Based on design specification, the system should be implemented using Java applet. Finally, we describe user manual as a guideline for using KWIC system. All the materials of project should be uploaded on our web site.

## Definitions, Acronyms, and Abbreviations

* KWIC: Key Word In Context describes a way to display related and accurate results to a specific search query.
* Architecture acronym if applicable
* UML(Unified Modeling Language): This provides a way to describe structure, behavior and architecture of application along with business process and data structure

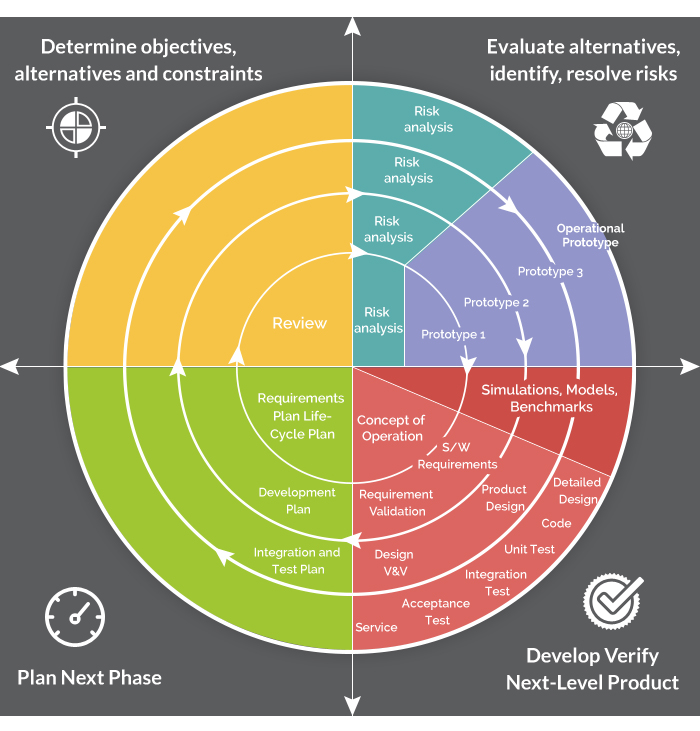
Phase 2b: Final Part II

Deliverables: Presentation and Demo  
 Due Date: December 1st

Team Leader: Twinkle Sharma

# PROJECT ORGANIZATION

## Process Model



## Organizational Structure

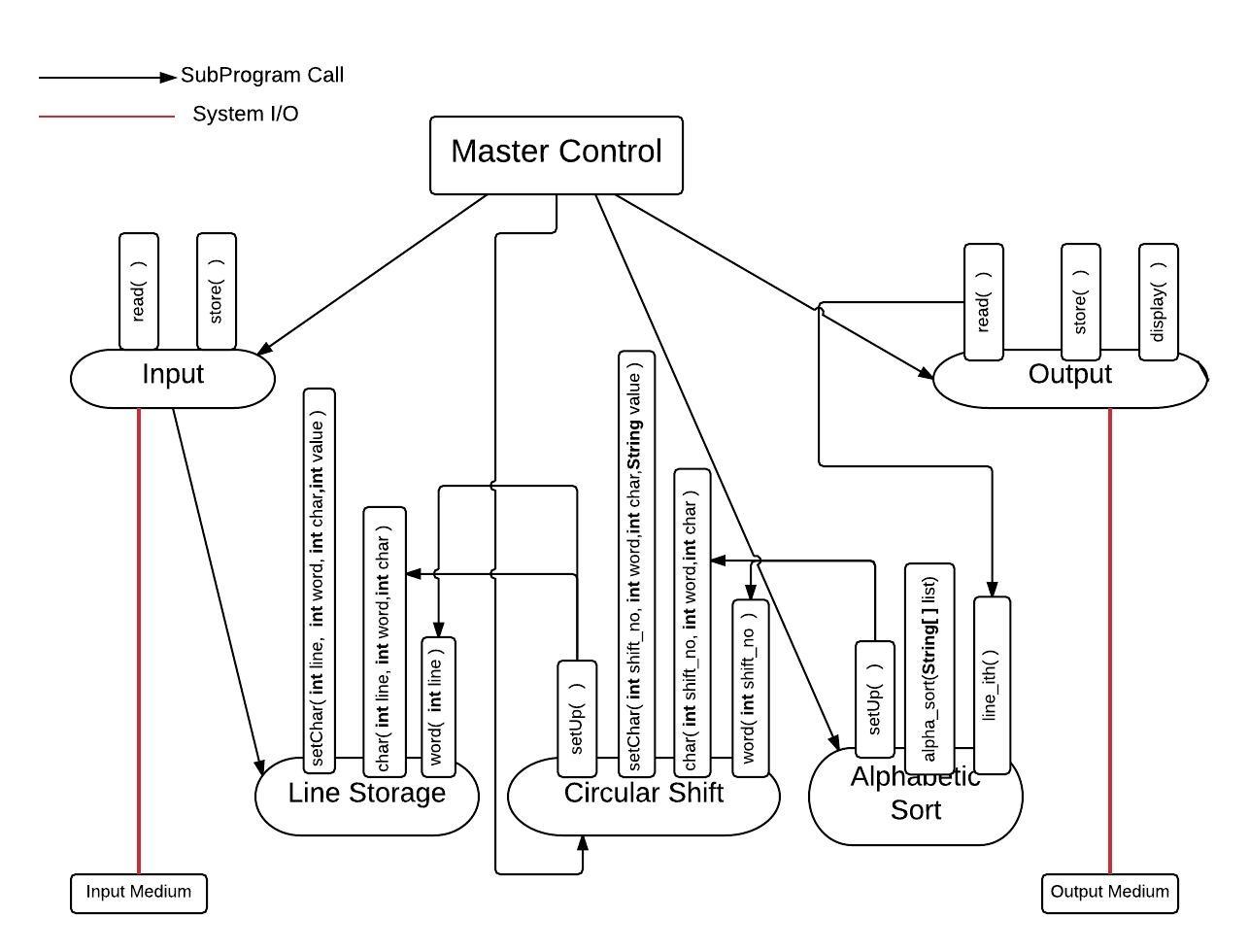
## Work Schedule

|  |  |  |
| --- | --- | --- |
| Date | Task | Responsibility |
| 9/03/2015 | Requirements Gathering | Sruthi, Twinkle, Maryellen, Barbara |
| 9/29/2015 | Software Architecture Analysis & Design | Sruthi, Barbara |
| 10/15/2015 | Developers | Twinkle, Maryellen |
| 11/10/2015 | Testers | Twinkle, Sruthi |
| 12/1/2015 | End users | Barbara, Maryellen |

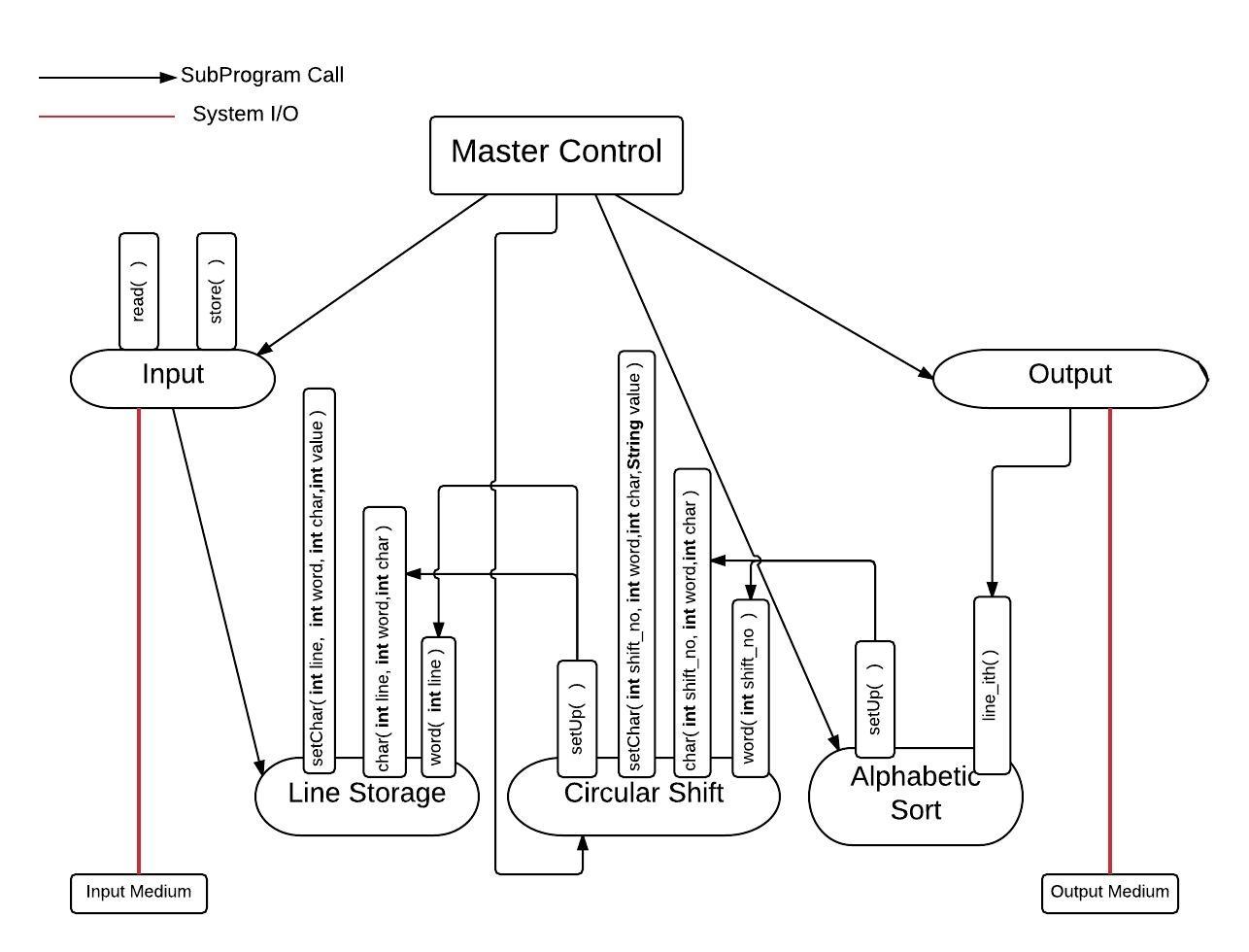
# Software Architecture

# **Abstract Data Type Architectural Style**

Design 1

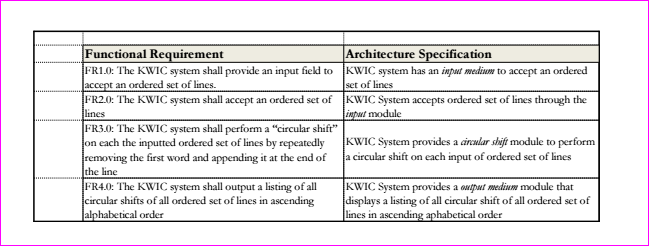


Design2:



# Traceability Matrix

## Functional Requirement and Software Architecture Traceability Matrix



# Prototype/Screen Shotskwic_screenshot3.png

# References

* http://www.utdallas.edu/~chung/SA/syllabus.htm
* *Documenting Software Architectures: Views and Beyond,*P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord and J. Stafford, MA: Addison-Wesley, 2003.