Software Requirements Specification

Version 1.0

April 27, 2023

Structured Analysis and Structured Design using CASE tools

Team Members:

Name	Roll no
Konchada Jyothikiran	2006063
Yallampati Hemavardhini	**2006074**
Anjali Kumari	2006077
Rida Khan	2006078
Gautam Babita	2006079
Bollina Kavyasri	2006085
Suddapu Sandeep	2006092
Divya Mishra	2006096
Kethavath Sundar	2006099
Para Venkata Akshith	20060108

Prepared for

CSL6402—Software Engineering Lab

Instructor: Prof. Anil Kumar sir.

1. Introduction

Perform structured analysis and structured design (SA/SD) for the following CASE tool for Structured Analysis Software to be developed for automating various activities associated with developing a CASE tool for structured software analysis.

1.1. Purpose

Development of automating various activities associated with developing a CASE tool for structured software analysis.

• The case tool should support a graphical interface and the following

features.

• The user should be able to draw bubbles, data stores, and entities and

connect them using data flow arrows. The data flow arrows are annotated

by the corresponding data names.

• Should support editing the data flow diagram.

• Should be able to create the diagram hierarchically. The user should be

able to determine balancing errors whenever required.

• The software should be able to create the data dictionary automatically.

• Should support printing the diagram on a variety of printers.

• Should support querying the data items and function names. The

diagrams matching the query should be shown.

1.2. Scope of Project

The scope of the project for performing structured analysis and structured design (SA/SD) is the tool will support editing the data flow diagram, creating

the diagram hierarchically, and determining balancing errors whenever required. The development process will include requirements gathering, data flow diagramming, data dictionary creation, testing, and implementation.

Overall, the scope of the project is to develop a CASE tool for structured software analysis that will enable users to automate various tasks, create diagrams hierarchically, and generate data dictionaries automatically, thereby streamlining the software analysis process.

1.3. Glossary

Term	Definition
SA/SD	Structured Analysis and Structured Design
CASE	Computer-Aided Software Engineering
Graphical Interface	A visual representation that allows users to interact with the software using icons, menus, and other graphic elements.
Bubbles	A graphical symbol used to represent a process or activity in a data flow diagram.
Data Stores	A graphical symbol used to represent a repository for data in a data flow diagram.
Software Requirements Specification	A document that describes what the software will do and how it will be expected to perform.
Entities	A graphical symbol used to represent an object or thing in a data flow diagram.
Data Flow Arrows	A graphical symbol used to represent the flow of data between bubbles, data stores, and entities in a data flow diagram.
Data Dictionary	A document that defines all data elements used in the system and their attributes.
Balancing Errors	Discrepancies between the input and output data flows in a data flow diagram.
Printing	The process of generating a hardcopy of the data flow diagram.
Querying	The process of searching for specific data items or function names in the data flow diagram.

1.4. Overview of Document

An overview is a brief summary or a general outline of a topic or a subject. In the context of your question, an overview of the given task would be to perform structured analysis and design for developing a CASE tool for automating various activities associated with developing a CASE tool for structured software analysis. The CASE tool should have a graphical user interface (GUI) and support the creation and editing of data flow diagrams. It should also be able to create the data dictionary automatically, support printing on different printers, and allow querying of data items and function names with the corresponding diagram shown. The software should support hierarchical creation of diagrams with balancing errors checked when required.

2. Software requirements specification

2a. Functional Requirements

The following are the functional requirements for the Structured Analysis Software CASE Tool:

• <u>Drawing Features</u>: The user should be able to draw bubbles, data stores, and entities and connect them using data flow arrows. The data flow arrows are annotated by the corresponding data names.

• <u>Editing Features</u>: The software should support editing the data flow diagram. The user should be able to modify the data flow diagram whenever required.

• <u>Hierarchical Diagram Creation</u>: The software should be able to create the diagram hierarchically. The user should be able to determine balancing errors whenever required.

• <u>Automatic Data Dictionary Creation</u>: The software should be able to create the data dictionary automatically.

• <u>Printing Features</u>: The software should support printing the diagram on a variety of printers.

• <u>Querying Features</u>: The software should support querying the data items and function names. The diagrams matching the query should be shown.

2b. Non-Functional requirements

• <u>Usability</u>: The system should have a user-friendly Interface that is easy to navigate. It should be intuitive and require minimal training for the book shop staff to use.

• <u>Performance</u>: The system should be able to handle a large volume of parallel transactions without any downtime and delay. It should be ready with real-time information of the sales data.

• <u>Security</u>: The system should ensure security of the customer data as well transaction related information and authentication keys by implementing authorization and access control hierarchies.

• <u>Reliability</u>: The tool should be reliable and stable, without significant bugs or errors.

• <u>Compatibility</u>: The tool should be compatible with various operating systems and browsers, and should be able to run on a range of hardware configurations

• <u>Maintainability</u>: The tool should be easy to maintain and update, with clear and well-documented code that can be easily understood and modified by developers.

3. Design

3a. Data Flow diagrams

A Data Flow Diagram (DFD) is a visual representation of how data flows through a system. It is a graphical tool used for depicting the flow of data and the processing of that data within a system. DFDs are commonly used in software engineering to model the structure and behavior of a software system.

A DFD consists of a set of interconnected bubbles, representing the different components of a system, and arrows, representing the flow of data between these components.
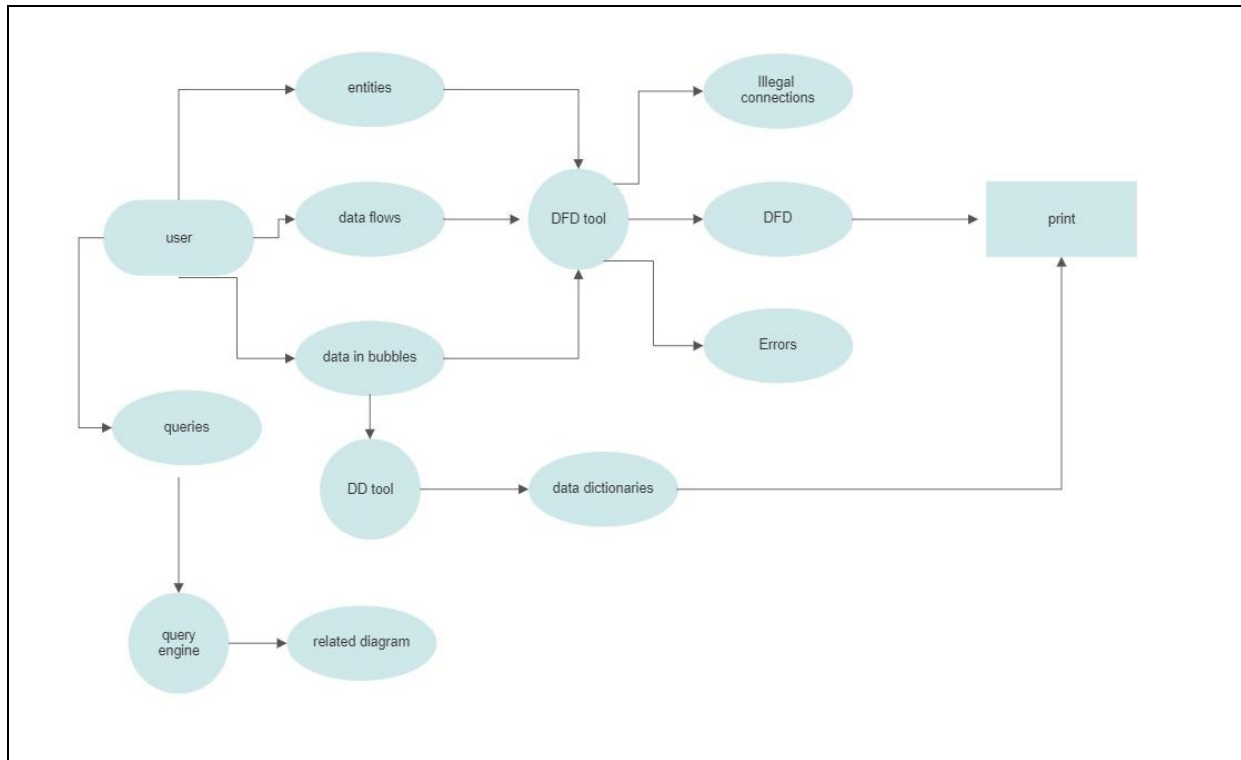
LEVEL 0 DFD:
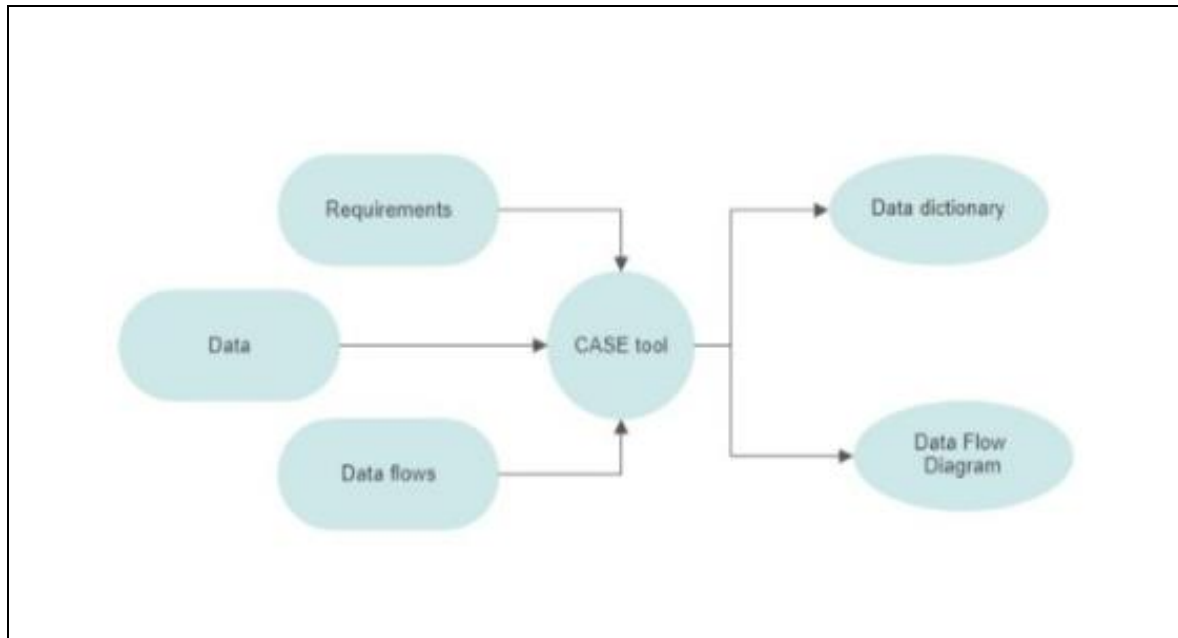
3a.1 Level-0-DFD

The above diagram is a 0-level DFD that only shows the main function of the system as bubbles and the data flows between them. The CASE tool has different functions including requirements, data, data flow, creating the data dictionary automatically and also editing the Data flow diagram.

LEVEL 1 DFD:

3a.2 Level-1-DFD

The above shown diagram is a 1-level Data Flow Diagram for Structured analysis and Structured Design using CASE tool. In above diagram user is the external entity that interacts with the CASE tool, includes the entities, data flows, data in bubbles, and querying the data items. In this diagram, the user interacts with the hierarchy module to create the diagram in a hierarchical structure and specify the balancing error if any. The hierarchical module then sends the hierarchy diagram and balancing error to the DFD tool and printing the diagram on a variety of printers
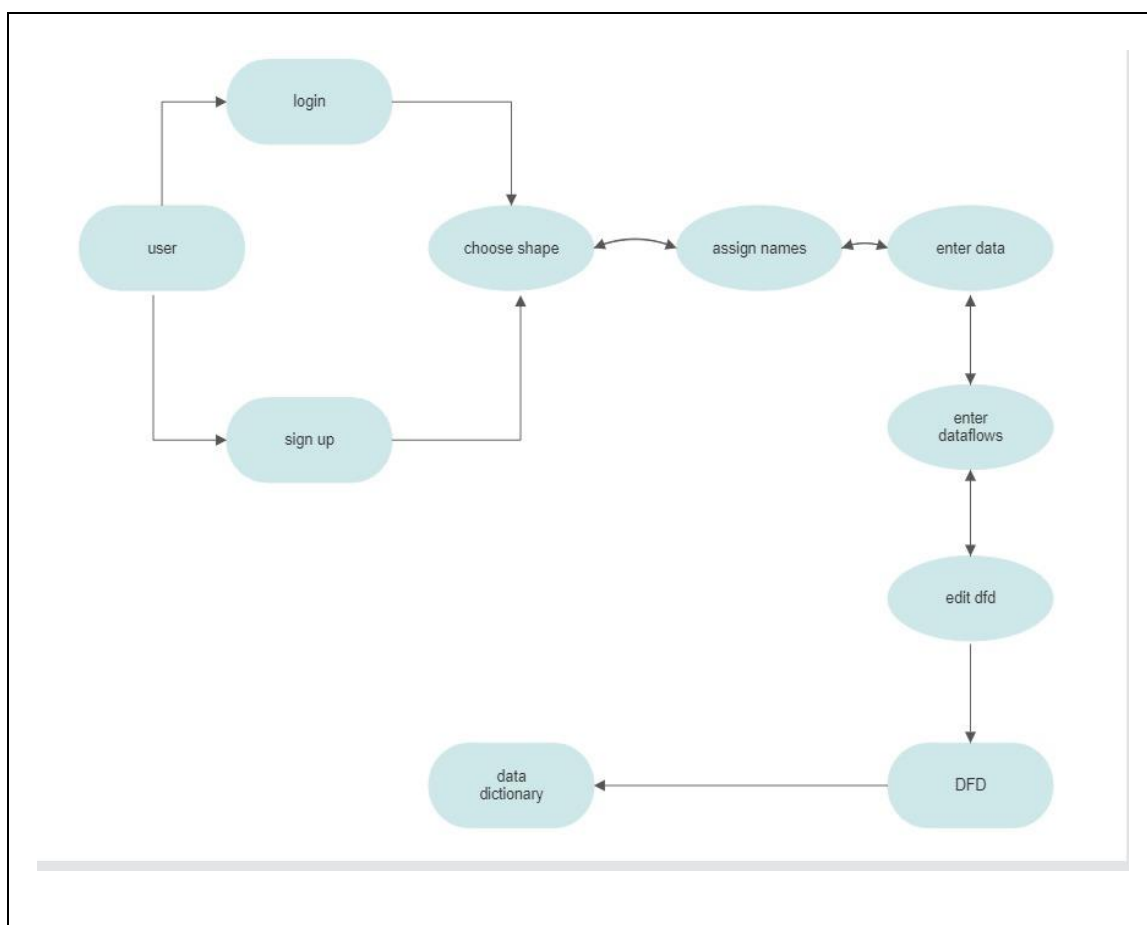
3b. Use Case diagrams

A use case diagram is a type of UML (Unified Modelling Language) diagram that depicts the interactions between the system under consideration and its users or external systems. A use case diagram typically consists of users, use cases, and the relationships between them.

This diagram represents that user done with the loin process then choosing shapes and assign names allows the user to draw bubbles, data stores and entities. The data flows are annotated by the corresponding data names.

Then allows the user to modify the data flow diagram by changing the components and connections. User is to create a diagram in a hierarchical structure, detecting balancing errors when required.

And automatic data dictionary creation happens, it automatically generates data dictionary entries.

3b Use case diagram