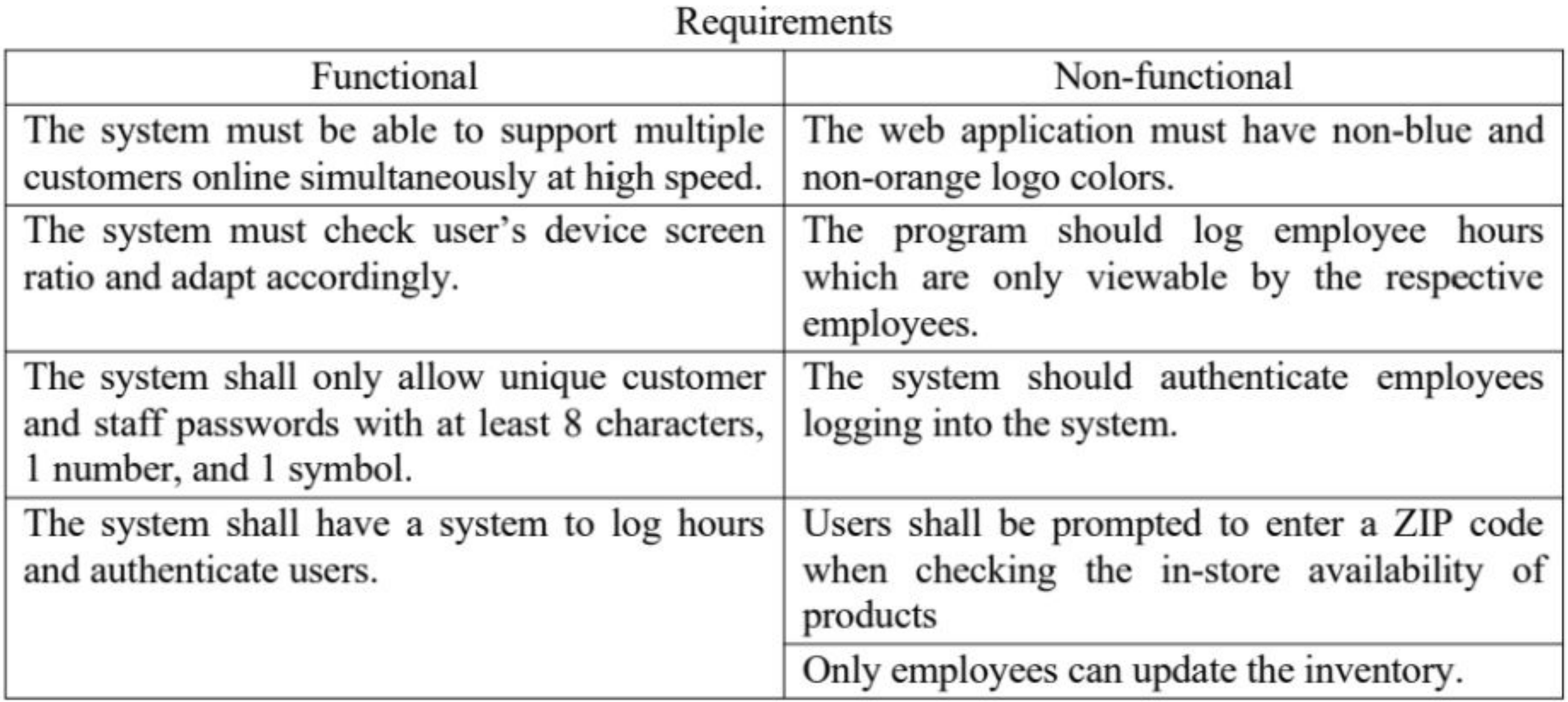
GRAND VALLEY STATE UNIVERSITY
www.gvsu.edu

Software Engineering Requirements Representations

Erik Fredericks // frederer@gvsu.edu

Adapted from materials provided by Byron DeVries, Jagadeesh Nandigam



Requirements

Functional	Non-functional
The system must be able to support multiple customers online simultaneously at high speed.	The web application must have non-blue and non-orange logo colors.
The system must check user's device screen ratio and adapt accordingly.	The program should log employee hours which are only viewable by the respective employees.
The system shall only allow unique customer and staff passwords with at least 8 characters, 1 number, and 1 symbol.	The system should authenticate employees logging into the system.
The system shall have a system to log hours and authenticate users.	Users shall be prompted to enter a ZIP code when checking the in-store availability of products
	Only employees can update the inventory.

Outline

Types of UML Diagrams

What is a System?

Structural Diagrams

Behavioral Diagrams

Diagram Development Process

Types of (UML) Diagrams

The objectives are to:

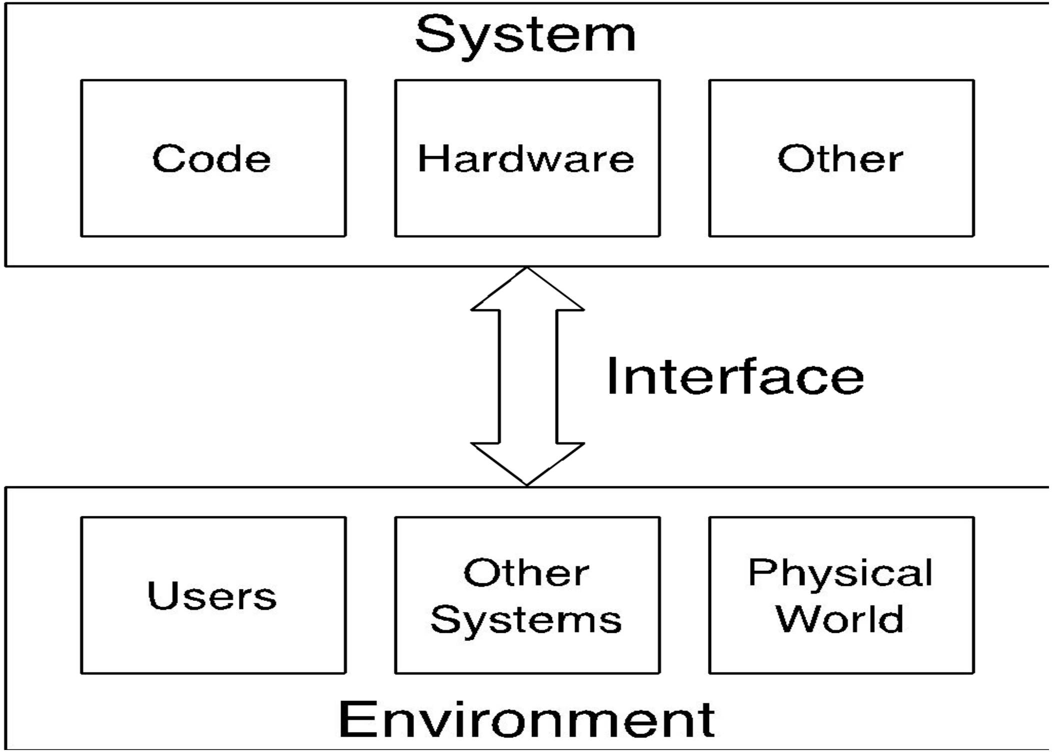
- Visualize,
- Specify,
- Construct, and
- Document a system.

Structural: Focus on the static aspects of a system

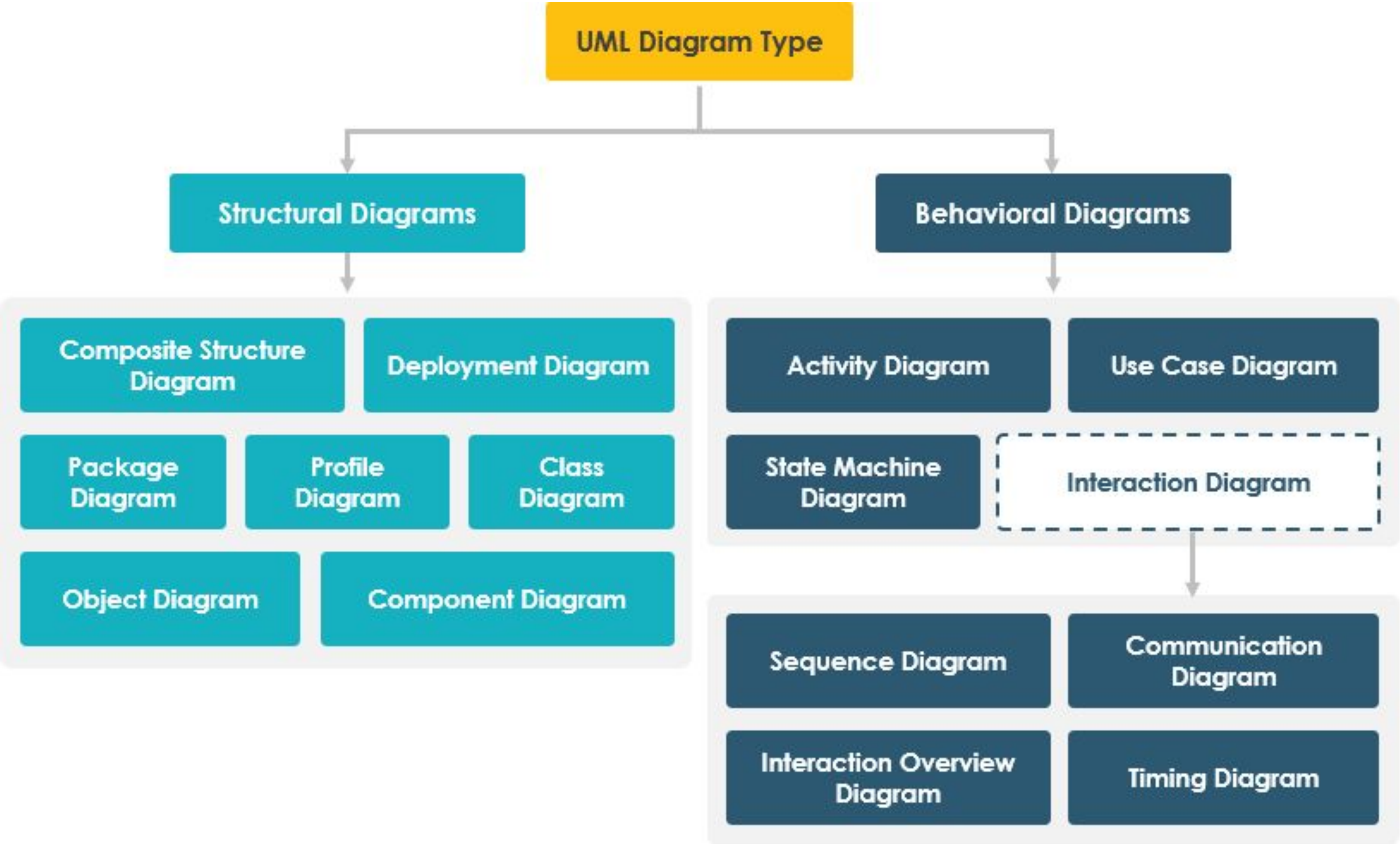
Behavioral: Focus on dynamic aspects of a system

But what is a system?

What is a System?



Where do we describe requirements?



UML Diagram Type

Structural Diagrams

Composite Structure Diagram

Deployment Diagram

Package Diagram

Profile Diagram

Class Diagram

Object Diagram

Component Diagram

Behavioral Diagrams

Activity Diagram

Use Case Diagram

State Machine Diagram

Interaction Diagram

Sequence Diagram

Communication Diagram

Interaction Overview Diagram

Timing Diagram

Structural Diagrams

Class: a set of classes and their relationships

- **Interface**: is a collection of operations that specify a service of a class

Object: Set of objects and their relationships

Component: Set of components and their relationships. The physical realization of logical groupings of elements (e.g., classes, interfaces)

Deployment: Set of nodes and their relationships

- Exists at run time; represents computational resource
- Node typically encloses one or more components

Behavioral Diagrams

Use-Case: Organize behaviors of the system

- User goals (high-level services of system)
- Perspective from external entities (actors)

Interaction Diagrams:

- Sequence: focus on time ordering of messages
- Collaboration: focus on structural organization of objects that send/receive messages

Statechart: changing state of the system by driving events

Activity: focus on flow of control from one activity to another

Diagram Development Process

High-Level Capture of Requirements

- Use-Case Diagram

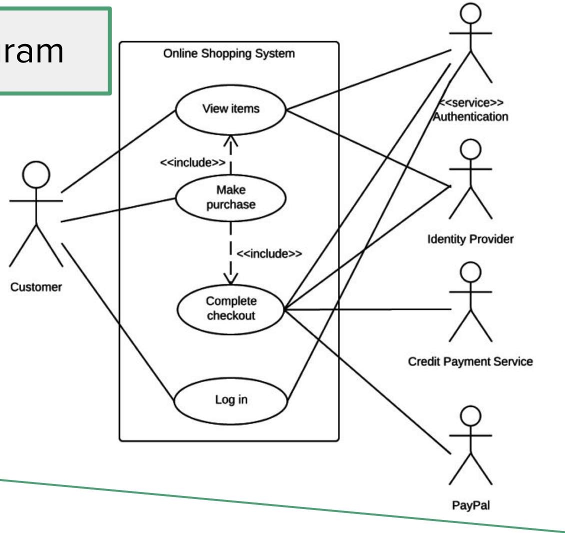
Identify major objects and relationships

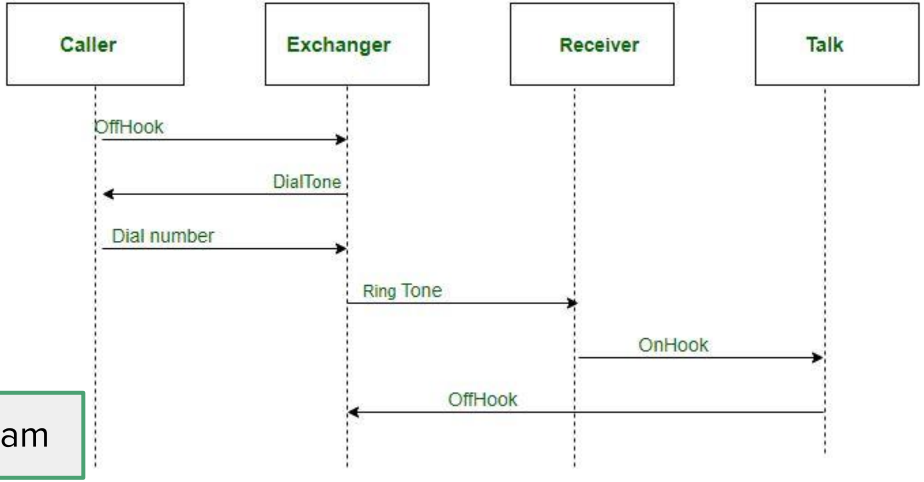
- Class diagram (object diagram)

Create scenarios of usage

- Interaction Diagrams
- Sequence Diagram
- Collaboration Diagram

Use case diagram





Interaction diagram

Diagram Development Process

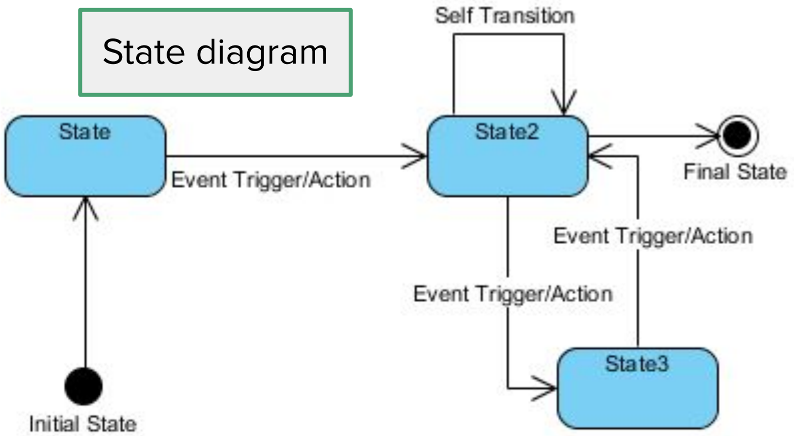
Generalize scenarios to describe behavior:

- State Diagram
- Activity Diagram

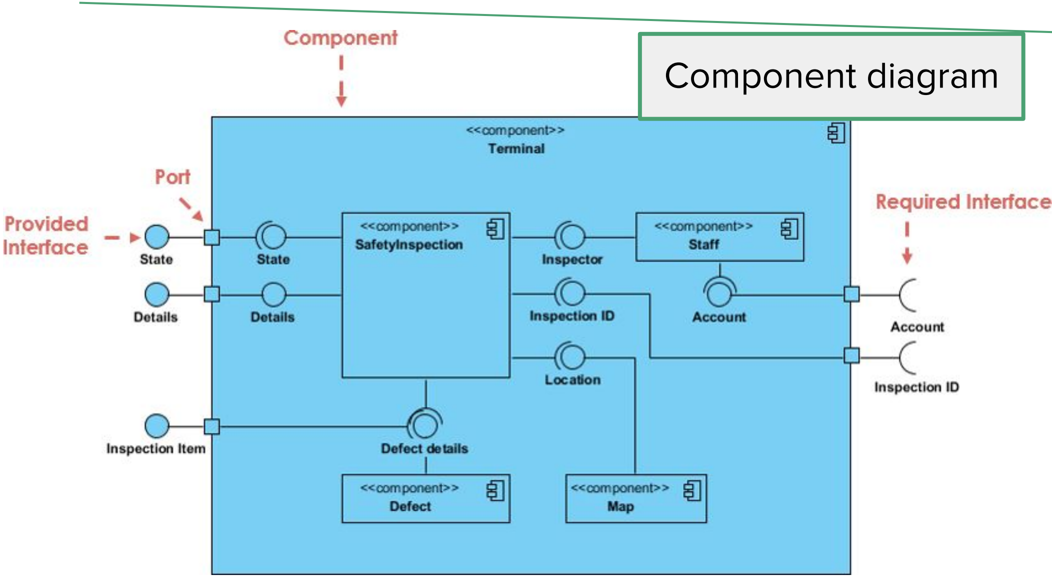
Refine to add implementation details:

- Implementation Diagrams
- Component Diagram
- Deployment Diagram

State diagram



Component diagram


Component
<<component>>
Terminal
Port
Provided Interface
State
State
<<component>>
SafetyInspection
Inspector
<<component>>
Staff
Required Interface
Details
Details
Inspection ID
Account
Account
Location
Inspection ID
Inspection Item
Defect details
<<component>>
Defect
<<component>>
Map