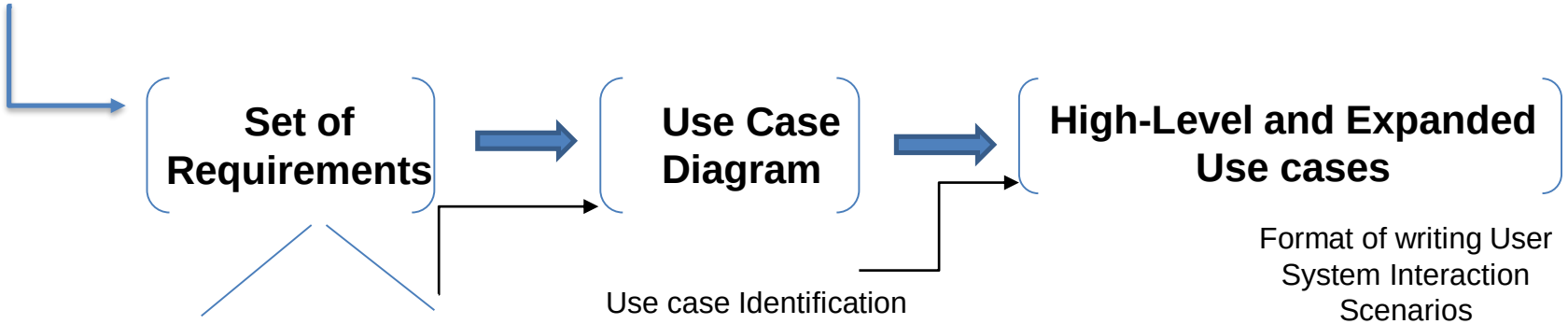
Software Design and Analysis
CS-3004
Lecture#04

Dr. Javaria Imtiaz,
Mr. Basharat Hussain,
Mr. Majid Hussain

Revision up till now

DOCUMENT (SRS)



Use case Identification

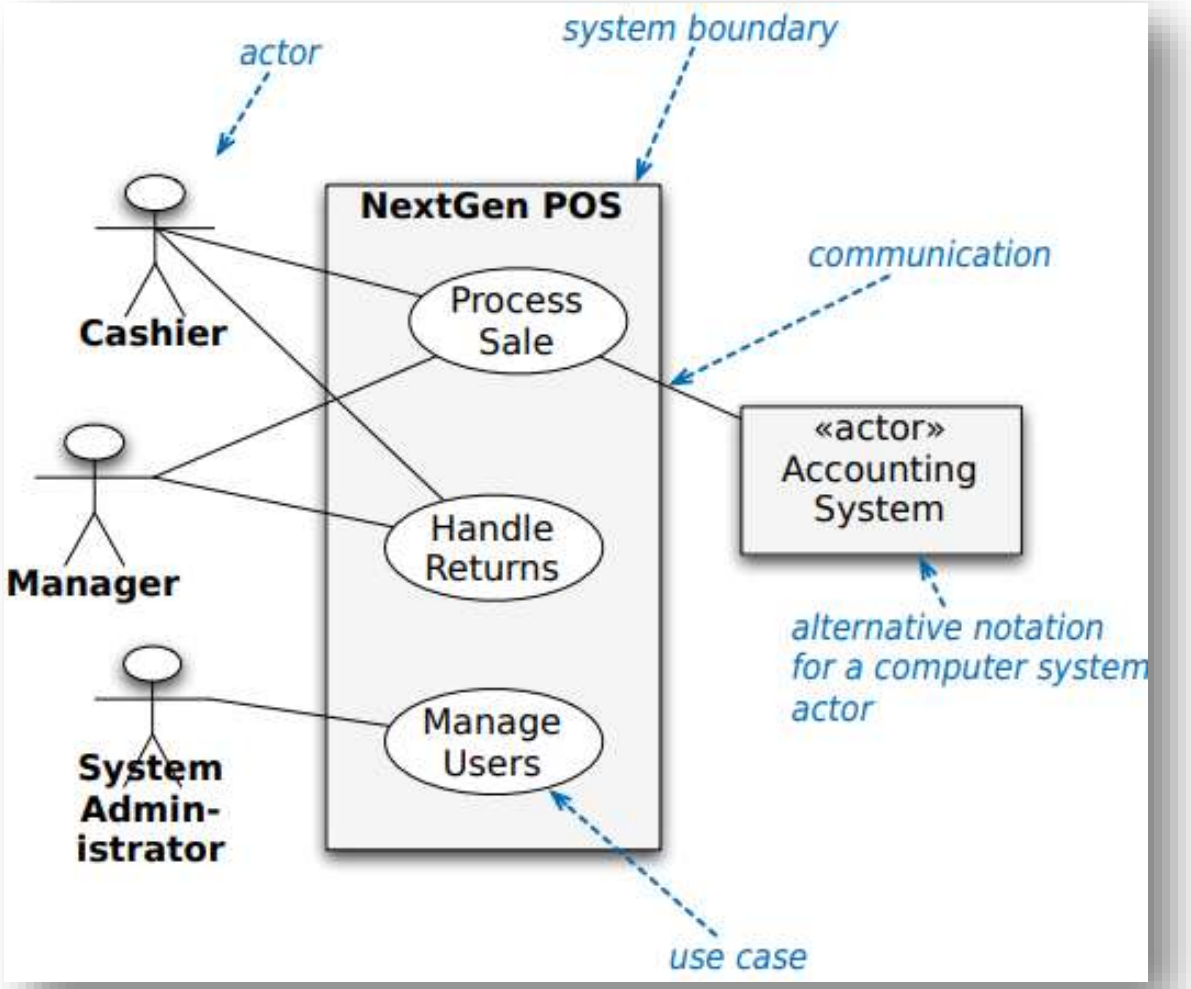
- Boss test
- Elementary Business Process
- Size Test

- ➢ Represent feature/functions/user goals
- ➢ Presented as VERB

Format of writing User System Interaction Scenarios

Recap

- UML use case diagrams provide a notation to illustrate the names of use cases and actors, and the relationship between them.



Use Case

- A use case is a set of scenarios tied together by a common user goal.
- Examples are:
  - “sign up for the exam”,
  - “make a bank transfer”.

Use Case Diagram

1. A use case diagram for the entire system – a graphic model.

2. Use cases are **not diagrams** they are **text documents.**

3. Use cases are one way of capturing the **functional requirements.**

4. Use cases are **text stories** of some **actors** using a system to **meet goals**.

5. One or more use case narratives for each use case – descriptions in text.
   - **High-level Use Cases**
   - **Expanded/Extended Use Cases**

Use Case

- What it is:
  - Text story
  - Widely used to discover and record (mostly functional) requirements

- What is it about:
  - Some actor(s) using a system to meet specific goals
  - Answering questions:
    - Who is using the system, what are their typical scenarios of use, and what are their goals?
- What it is NOT:
  - Not object-oriented
  - Not a diagram
    - UML use cases diagrams are “secondary-value” artifacts
- Focus: use cases, not use case diagrams

Withdraw Money through ATM?
Submit Online Exam
Place an order Food Panda

Use Case Scenario

- A scenario is a specific sequence of actions and interactions between actors and the system under discussion.
- It is also called a use case instance.
- It is one particular story of using a system, or one path through the use case; for example, the scenario of successfully purchasing items with cash, or the scenario of failing to purchase items because of a credit card transaction denial.

Process Sale:

1. A customer arrives at a checkout with items to purchase.
2. The cashier uses the POS system to record each purchased item.
3. The system presents a running total and line-item details.
4. The customer enters payment information, which the system validates and records.
5. The system updates inventory.
6. The customer receives a receipt from the system and then leaves with the items.

Actors, Scenarios, and Use Cases

- **Actor**: entity that shows a behavior,
  - e.g.: a person (role), computer system, or organization
- **Scenario:** specific sequence of actions and interactions between actors and a system
  - use case instance
  - singe path of using the system
  - e.g., purchasing 10 items with cash (or even more detailed)
- **Use case:** collection of related success & failure scenarios that describe an actor using a system to support a **goal**

Use Case Example with Scenarios (casual format)

- **UC Handle Returns**
- *Main success Scenario*: A customer arrives at a checkout with items to return. The cashier uses the POS system to record each returned item …
- *Alternate Scenarios*:
  - if the customer paid by credit ...
  - If the item identifier is not found in the system …
  - If the system detects failure to communicate with the external accounting system …

Three Kinds of Actors

- Primary actor
  - has user goals fulfilled through using services of the system under discussion
  - drives the use cases
- Supporting actor
  - provides a service to the system under discussion
  - e.g., payment authorization service
    - implies: clarification of external interfaces and protocols needed
- Offstage actor
  - has an interest in the behavior of the use case, but is not primary or supporting
  - e.g., a government tax agency

Use Case Format

- **Brief**
  - Succinct one-paragraph summary usually the
  - main success scenario
  - done during early requirements analysis should take
  - only a couple of minutes
- **Casual**
  - informal paragraph format
  - multiple paragraphs covering various scenarios
- **Fully dressed**
  - details all steps and variations
  - includes supporting sections such as preconditions and success guarantees
  - mainly done after many use cases are identified and during early requirements workshop for high-value and high-risk requirements (e.g., core architectural)

High Level/Brief Use Case Format

Example: High-Level Use Case: Buy Items	
Use case:	Buy Items
Actors:	Customers?, Cashiers
Type:	Primary
Description:	A Customer arrives at a checkout with items to purchase. The cashier records the purchase items and collects payment. On completion, the Customer leaves with the items.

A brief description of use case's role and purpose

High Level/Brief Use Cases

UC01: Use case: Buy Items
Actors: Customer (initiator), Cashier
Type: primary
Description: A Customer arrives at a checkout with items to purchase. The Cashier records the purchase items and collects a payment. On completion, the Customer leaves with the items.

UC02:Use case: Start Up
Actors: Manager
Type: primary
Description: A Manager powers on a POST in order to prepare it for use by Cashiers. The Manager validates that the date and time are correct, after which the system is ready for Cashier use.

A Template for Fully Dressed Style

- **Use case name**
  - start with a verb
- **Scope**
  - the system under design
- **Level**
  - user goal or subfunction level
- **Primary actor**
  - calls on the system to deliver a service
- **Stakeholders and interests**
  - who cares about this use case, and what they want?
- **Preconditions**
  - what must be true on start, and worth telling the reader
- **Success guarantee (postcondition)**
  - what must be true on successful completion, worth telling the reader
- **Main success scenario**
  - a typical, unconditional happy path scenario of success
- **Extensions**
  - alternate scenarios of success and failure
- **Special requirements**
  - related non-functional requirements
- **Technology and data variations list**
  - varying I/O methods and date formats

Scope

- Defines how broad the use case is.
- This can be for the whole system, as in the POS example, or narrow, as in a use case for creating a journal entry in an accounting system.

Level

- User-goal: Scenarios that let a user get something done. Corresponds to an elementary business process.
- Sub function: smaller steps required to support a user goal.

user-level for a use case that describes one complete activity in the system; or ***subfunction*** for a use case that depends on a user-level use case but is too long to include in the user-level use case.

Primary Actor

- The person (or sometimes object) that calls upon system services to fulfill a goal. (When might an actor not be a person?)

Stakeholders and Interests

- The stakeholders are people who have a reason to want this system. The Interests are their reasons for wanting it and what they expect from it.
- You could view the system as a contract between various stakeholders.

Someone or something with an interest in the behavior of the system under discussion. E.g. company stakeholders, customers, vendors, and government regulatory agencies,…

Preconditions

- It announces what the system will ensure is true before letting the use case start. Since it is enforced by the system and known to be true, it will not be checked again during the use case execution; e.g. user has logged in.

Success Guarantee

- States what interests of the stakeholders are satisfied after a successful conclusion of the use case, either at the end of the main success scenario or at the end of a successful alternative path. The success guarantees are written additionally to the minimal guarantees.

Main Success Scenario

- This satisfies the interests of the stakeholders. You get your groceries, the store gets your money, inventory is reduced, etc.
- Steps:
  - An interaction between actors
  - Validation (by the system)
  - State change to the system

a case in which nothing goes wrong.

One Column Format

Success Guarantee (or Postconditions): Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.

Main Success Scenario (or Basic Flow):

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.

Cashier repeats steps 3-4 until indicates done.

5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods (if any).

Two Column Format

Main Success Scenario:

Actor Action (or Intention)	System Responsibility
1.Customer arrives at a POS checkout with goods and/or services to purchase.	
2.Cashier starts a new sale.	
3.Cashier enters item identifier.	4.Records each sale line item and presents item description and running total.
Cashier repeats steps 3-4 until indicates done.	5.Presents total with taxes calculated.
6.Cashier tells Customer the total, and asks for payment.	
7.Customer pays.	8.Handles payment.
	9.Logs the completed sale and sends information to the external accounting (for all accounting and commissions) and inventory systems (to update inventory). System presents receipt.

Two column emphasizes interaction

Extensions or Alternate Flows

- These include all other possible outcomes, both success and failure.

what can happen differently during a scenario.

Technology and Data Variations

- Technical variations on how something must be done:
  - Scan bar code
  - Key item ID
  - RFID
- Avoid early design decisions; keep things general.

Excerpt of a Fully Dressed Use Cases

Name:	Buy Stocks over the Web
Primary Actor:	Purchaser
Scope:	Finance Package (PAF)
Level:	User goal

Stakeholders and Interests:

Purchaser - wants to buy stocks and get them added to the portfolio.
Stock agency - wants full purchase information.

Precondition:

User is logged in.

Success guarantee:

Web site has acknowledged the purchase; the logs and the user's portfolio are updated.

Cont..

Main Success Scenario:

1. Purchaser selects to buy stocks over the web
2. PAF gets name of web site to use (A, B,…) from user
3. PAF opens web connection to site, retaining control
4. Purchaser browses and buys stock from the web site
5. PAF intercepts responses from the web site and updates purchaser's portfolio
6. PAF shows the user the new portfolio standing

Cont..

Extensions:

2a. Purchaser wants a web site PAF does not support

2a1. System gets new suggestion from purchaser, with option to cancel

4a. Web site does not acknowledge purchase, but puts it on delay

4a1. PAF logs the delay, sets a timer to ask the purchaser about the outcome

Use Case UC1: Process Sale

Scope: NextGen POS application

Level: user goal

Primary Actor: Cashier

Stakeholders and Interests:

- Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.

- Salesperson: Wants sales commissions updated.

- Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.

- Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server components (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.

- Manager: Wants to be able to quickly perform override operations, and easily debug Cashier problems.

- Government Tax Agencies: Want to collect tax from every sale. May be multiple agencies, such as national, state, and county.

- Payment Authorization Service: Wants to receive digital authorization requests in the correct format and protocol. Wants to accurately account for their payables to the store.

Preconditions: Cashier is identified and authenticated.

Success Guarantee (or Postconditions): Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.

Success Guarantee (or Postconditions): Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.

Main Success Scenario (or Basic Flow):

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.

Cashier repeats steps 3-4 until indicates done.

5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods (if any).

1. Cashier restarts System, logs in, and requests recovery of prior state.

2. System reconstructs prior state.

    2a. System detects anomalies preventing recovery:

        1. System signals error to the Cashier, records the error, and enters a clean state.

        2. Cashier starts a new sale.

3a. Invalid item ID (not found in system):

    1. System signals error and rejects entry.

    2. Cashier responds to the error:

        2a. There is a human-readable item ID (e.g., a numeric UPC):

            1. Cashier manually enters the item ID.

            2. System displays description and price.

                2a. Invalid item ID: System signals error. Cashier tries alternate method.

4a. The system supplied item price is not wanted (e.g., Customer complained about something and is offered a lower price):

1. Cashier requests approval from Manager.

2. Manager performs override operation.

3. Cashier enters manual override price.

4. System presents new price.

5a. System detects failure to communicate with external tax calculation system service:

6a. Customer says they intended to pay by cash but don't have enough cash:

1. Cashier asks for alternate payment method.

    1a. Customer tells Cashier to cancel sale. Cashier cancels sale on System.

Special Requirements:

- Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.

- Credit authorization response within 30 seconds 90% of the time.

- Somehow, we want robust recovery when access to remote services such the inventory system is failing.

- Language internationalization on the text displayed.

- Pluggable business rules to be insertable at steps 3 and 7.

Technology and Data Variations List:

*a. Manager override entered by swiping an override card through a card reader, or entering an authorization code via the keyboard.

3a. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.

3b. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.

7a. Credit account information entered by card reader or keyboard.

7b. Credit payment signature captured on paper receipt. But within two years, we predict many customers will want digital signature capture.

Write in a UI-Free Style

- Most programs are dependent upon a particular user interface. However, avoid constraining your program too early:
- “The user keys an ID and password into a dialog box and presses the OK button.”
- “The user identifies himself to the system.”
- The latter allows for biometric ID, keyin, etc.

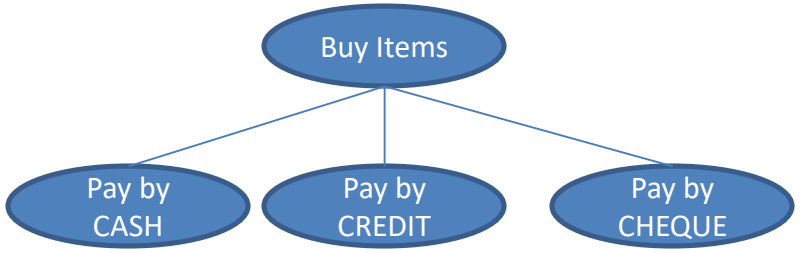
Essential Style

- Focus on the essence, or basic idea, not the details of implementation
- Contrast with concrete style

Write Black-Box Use Cases

- Focus on what the system must do,
  - i.e., the behavior or functional requirements
  - Not on how it will do (the design)
- Examples:
  - Good: The system records the sale
  - Bad: The system writes the sale to the database.
  - Worse: System generates SQL INSERT statement for the sale

Expanded Use Cases



Section Main	
Use Case	Buy Items
Actors	Customer (initiator), Cashier
Purpose	Capture a sale and its payment.
Overview	A Customer arrives at a checkout with items to purchase. The Cashier records the purchase items and collects a payment. On completion, the Customer leaves with the items.
Type	primary and essential

Expanded Essential Use Cases (2)

Typical Course of Events	
Actor Action	**System Response**
1. This use case begins when Customer arrives at the POST checkout with items to purchase.	
2. The Cashier records each item. If there is more than one item, Cashier can enter the quantity as well.	3. Determines the item price and adds the item information to the running sales transaction. The description and price of the current item are presented.
4. On completion of item entry, the Cashier indicates to the POST that item entry is complete.	5. Calculates and presents the sale total.
6. The Cashier tells the Customer the total.	
7. Customer chooses payment type: a. If cash payment, see **section Pay by Cash**. b. If credit payment, see **section Pay by Credit**. c. If check payment, see **section Pay by Check**	8. Logs the completed sale. 9. Updates inventory levels. 10. Generates a receipt.

Expanded Essential Use Cases (3)

Typical Course of Events	
Actor Action	**System Response**
11.The Cashier gives the receipt to the Customer.	
12.The Customer leaves with the items purchased.	
Alternative Courses Line 2: Invalid item identifier entered. Indicate error. Line 7: Customer could not pay. Cancel sale transaction.	

Expanded Essential Use Cases (4)

Section: Pay By Cash	
Actor Action	System Response
1. The Customer gives a cash payment-the "cash tendered" - possibly greater then the sale total.	
2. The Cashier records the cash tendered.	3. Shows the balance due back to the Customer.
4. The Cashier deposits the cash received and extracts the balance owing. The Cashier gives the balance owing to the Customer.	
Alternative Courses **Line 1:**Customer does not have sufficient cash. May cancel sale or initiate another payment method. **Line 4:** Cash drawer does not contain sufficient cash. Cashier requests additional cash from supervisor or asks Customer for different payment amount or method.	

Expanded Essential Use Cases (5)

Section: Pay By Credit	
Actor Action	**System Response**
1. The Customer communicates their credit information for the credit payment.	2.Generates a credit payment Request and sends it to an external Credit Authorization service.
3. Credit Authorization Service authorizes the payment	4. Receives a credit approval Reply from the Credit Authorization Service (CAS)
	5. POST (records) the credit payment and approval reply information to the Accounts Receivable system.(The CAS owes money to the Store, hence Acct/Recv must track it).
	6. Display authorization success message.
Alternative Courses **Line 3:** Credit request denied by Credit Authorization Service. Suggest different payment method.	

Expanded Essential Use Cases (6)

Section: Pay by Check

Actor Action	System Response
1. The Customer writes a check and identifies him/her-self.	
2. Cashier records the identification information and requests check payment authorization.	3. Generate a check payment request and sends it to an external Check Authorization Service.
4. Check Authorization Service authorizes the payment.	5. Receives a check approval reply from the Check Authorization Service.
	6. Indicates authorization success.

Alternative Courses

* Line 4: Check request denied by Check Authorization Service. Suggest different payment method.

EXAMPLE: Use Case. (Brief)

- Here's one in a *brief format*:

  – **Rent Videos**. A Customer arrives with videos to rent. The Clerk provided the required information. The System outputs information on each. The Clerk requests the rental report. The System outputs it, which is given to the Customer with their videos.

EXAMPLE: Fully Dressed

Use Case UC1: Rent Video
Level: User-level goal
Primary Actor: Clerk
Preconditions:

- Clerk is identified and authenticated.

Stakeholders and their Interests:
Clerk: Wants accurate, fast entry.
Customer: Wants videos, and fast service with minimal effort.
Accountant: Wants to accurately record transactions.
Marketing: Wants to track customer habits.
. . .

EXAMPLE: Fully Dressed

***Main Success Scenario (or Basic Flow or "Happy Path")*:**

1. Customer arrives at a checkout with videos or games to rent.
2. Clerk enters Customer ID.
3. Clerk enters rental identifier.
4. System records rental line item and presents item description.
   (Clerk repeats steps 3-4 until indicates done.)
5. System displays total.
6. Customer pays. System handles payment.
7. Clerk requests rental report.
8. System outputs it. Clerk gives it to Customer.
9. Customer leaves with rentals and report.

EXAMPLE: Fully Dressed

***Extensions (or Alternatives)*:**

a*. At any time, System fails:

1. Clerk restarts System
2. logs in
3. requests recovery from prior state

1a. New Customer.

1. Perform use case Manage Membership.

2a. Customer ID not found.

1. Cashier verifies ID. If entry error, reenter, else Manage Membership.

2b. Customer has unpaid fines (usually for late returns).

1. Pay Fines.

EXAMPLE: Fully Dressed

Special Requirements:

- Language internationalization on the display messages and rental report.
- Large text on display. Visible from 1 m.

Technology and Data Variations:

- ID entries by bar code scanner or keyboard.