

# Software Requirements Specification

## Example

**Project Name: PruneIt**

*2019 B.A.N.D. Corporation*

### Revision History

Date	Revision	Description	Author
01/30/2019	1.0	Initial Version	Tony Romo

## Contents

Select table of contents below, then use menu **Insert | Reference | Index and Tables** to refresh it. This TOC will only show headings created using the built-in Word styles: Heading 1, Heading 2, and Heading 3.

<u>1.</u>	<u>Introduction</u>	1
1.1	<u>Purpose</u>	1
1.2	<u>Scope</u>	1
1.3	<u>References</u>	1
1.4	<u>Assumptions and Dependencies</u>	1
<u>2.</u>	<u>Software Product Overview</u>	2
2.1	<u>System Scope</u>	2
2.2	<u>System Architecture</u>	2
2.2.1	<u>External View of Software Product</u>	2
2.2.2	<u>Internal View of Software Product</u>	2
2.3	<u>Feature Overview</u>	2
<u>3.</u>	<u>System Use</u>	3
3.1	<u>Support For User Workflows</u>	3
3.2	<u>Actor Survey</u>	3
3.2	<u>Use-Case Model and System Events</u>	3
3.3	<u>System Interfaces</u>	4
<u>4.</u>	<u>Specific requirements</u>	5
3.1	<u>System Use-Cases</u>	5
	<u>Use Case 1: &lt;Insert Use-Case Name here&gt;</u>	5
	<u>Use Case 2: &lt;Insert Use-Case Name here&gt;</u>	6
	<u>Use Case n: &lt;Insert Use-Case Name here&gt;</u>	6
4.2	<u>System Functional Specification</u>	6
4.3	<u>System Domain Models</u>	6
4.4.1	<u>Internal Domain Model</u>	6
4.4.2	<u>Data Models</u>	7
4.4	<u>Non-Functional Requirements</u>	7
4.4.1	<u>Usability</u>	7
4.4.2	<u>Reliability</u>	7
4.4.3	<u>Performance</u>	8
4.4.4	<u>Supportability</u>	8

<u>5.</u>	<u>SUPPLEMENTARY REQUIREMENTS</u>	8
<u>5.1</u>	<u>Project management strategy</u>	8
<u>5.2</u>	<u>Systems Security and Audit</u>	8
<u>5.3</u>	<u>Assumptions and Dependencies</u>	8
<u>5.4</u>	<u>Requirements Traceability</u>	8
<u>6.</u>	<u>Online User Documentation and Help System Requirements</u>	9
<u>7.</u>	<u>Design Constraints</u>	9
<u>8.</u>	<u>Purchased Components</u>	10
<u>9.</u>	<u>Interfaces</u>	10
<u>9.1</u>	<u>User Interfaces</u>	10
<u>9.2</u>	<u>Hardware Interfaces</u>	10
<u>9.3</u>	<u>Software Interfaces</u>	10
<u>9.4</u>	<u>Communications Interfaces</u>	10
<u>10.</u>	<u>Licensing Requirements</u>	10
<u>11.</u>	<u>Legal, Copyright, and Other Notices</u>	10
<u>12.</u>	<u>Applicable Standards</u>	11
<u>Glossary</u>		11
<u>Appendixes</u>		11

# **1. Introduction**

## **1.1 Purpose**

This document is intended to fully specify the requirements and detail the software architecture of PruneIt. This Software Requirements Specification (SRS) will describe the external behavior of the application as well as the internal workings of the system. It will also describe the non-functional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements that were gathered and described in the Stakeholder Analysis, Problem/Context Analysis and Vision documents.

PruneIt will be developed with an iterative approach as outlined in the Rational Unified Process (RUP). Therefore, this specification will be a living/breathing document throughout the course of the project's Software Development Life Cycle (SDLC). As such, any section of this document is subject to change based on the feedback and outcomes of each implementation sprint.

## **1.2. Scope**

PruneIt is an application to be built for our landscaping clients. They have expressed the need to automate many of their business processes such as invoicing and scheduling. They would also like to have all of their data available electronically to better manage the current and future job requests as well as maintain a proper record of the past. Combining all of these features will better enable the landscaping companies to serve the customer by providing more accurate and efficient invoicing options as well providing better services as a result of the improved job scheduling system. Additionally, the landscaping companies hope to reduce the amount of time that is spent on manual administrative tasks and wish to have a better means of forecasting financial data by viewing reports on current and past transactions. The integration of all of these modules will allow company executives to clearly see what the company is doing well and what could be improved upon.

PruneIt will be developed in three major phases. The first phase will focus on the creation of the primary infrastructure or backbone of the application. In this phase, the database and web server will be configured and populated with the application structure. The initial data load will be done and the web framework will be primed with the main content areas of the site. All of the procurement and configuration of the tools needed will be completed in this phase. In phase two, the core web application modules will be built. After completion of this phase, eighty percent of the final solution will be ready and some of our clients may want or need to begin using the production system. Phase 3 will focus on the secondary use cases such as supplemental reports, advanced scheduling and time tracking features as well as mobile web application integration.

## **1.3 Assumptions and Dependencies**

A.) Given the agile nature of our proposed development schedule, it is assumed that the landscaping clients will be willing and able to help throughout the project's SDLC.

B.) B.A.N.D. will work with the client to do the initial data load of their current customer and job data into the database; however, subsequent additions must be made through the user interface by a member of the landscaping company.

C.) Because this application is web-based, the entire system will be available on mobile devices that have a web browser; however, only a subset of the content will be customized as "mobile"

specific content.

D.) Upon purchase of the software, managers of the client companies will be given an administrative account that has the ability to create other accounts with the company; after the initial set up process has been completed, all new users will be created by company management through the application's user interface.

## **2. Software Product Overview**

This section provides an overview of PruneIt as a software product. It contains a summary of how the system works as a whole from all user perspectives. Included in this section is the system scope, the system architecture, and an overview of the system features that make up PruneIt.

### **2.1 System Scope**

This section provides the physical (architectural) system scope of PruneIt. PruneIt is a standalone system that is intended for assisting in the management of landscaping businesses. Rather than the cumbersome and unorganized paper management process that currently exist; PruneIt will automate most of these tasks and thus assist landscaping businesses in being more productive, organized and efficient.

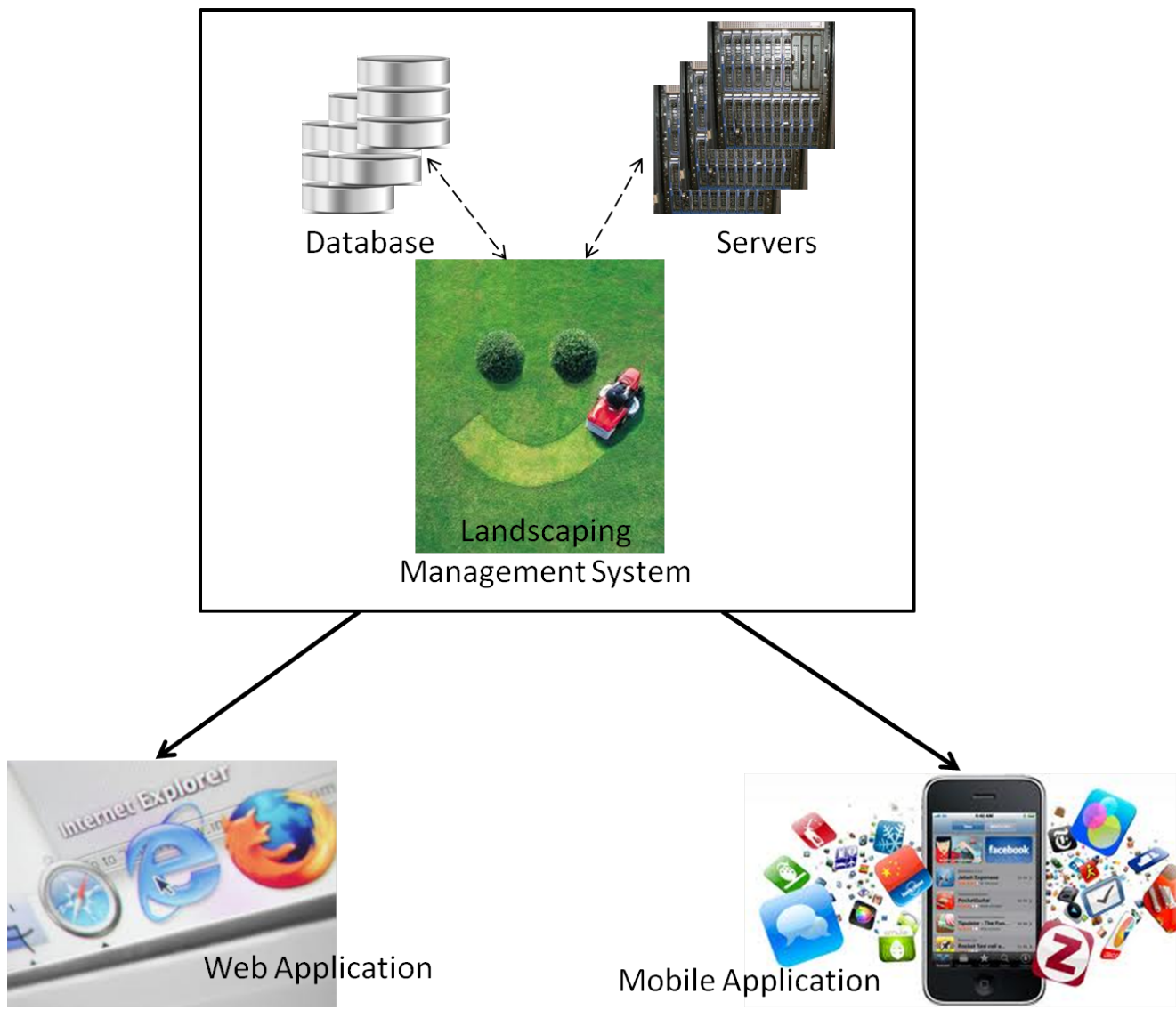
PruneIt will be available via both web and mobile applications. The web application will contain every component of the system where the mobile app will consist of a smaller set of components for landscapers on the fly. The web app will contain all the features needed by the company manager/owner. This includes the customer and employee storage, job schedule, equipment management, time tracking, report generation, job tracking and billing, and job type storage.

The mobile app is more intended for the "on the go" landscaper. It will contain all the necessary components needed for tracking time and jobs on the fly. This includes time and job tracking, expense input, and the viewing and updating of equipment inventory.

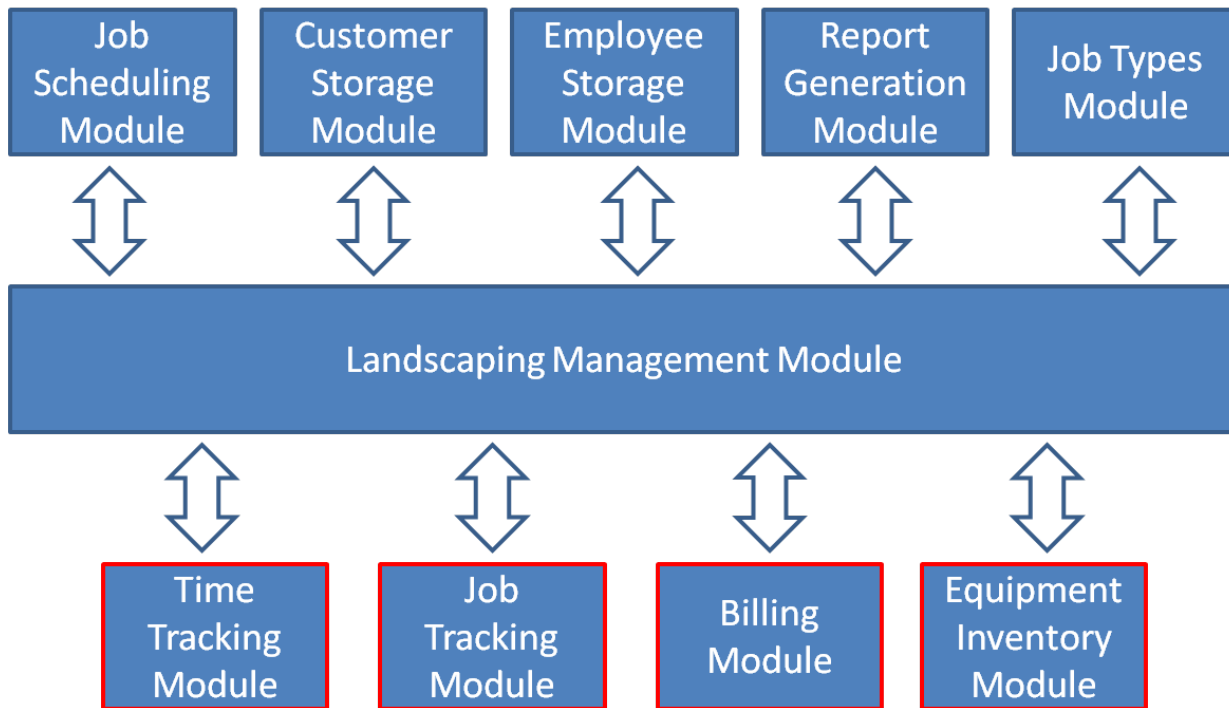
### **2.2 System Architecture**

This section defines the internal and external system architecture of PruneIt.

#### **2.2.1 External View of Software Product**



### 2.2.2 Internal View of Software Product



### 2.3 Feature Overview

This section provides a high-level overview of the system features available through PruneIt. The features below are subdivided into user needs from the landscaping owners and landscaping workers perspective.

Landscaping Owner/Manager (only available via the web application)

- F1: Input and Manage Customer Information – This feature allows users to store all customer information such as name, address, telephone, email as well as information about the customer's landscaping needs.
- F2: Input and Manage Customer Information - This feature allows users to store all employee information such as name, address, telephone, email as well as information on employee availability
- F3: Input and Manage Equipment Information - This feature allows users to store all equipment information such as status, type, where purchased, price.
- F4: Generate Reports – Users will be able to create company reports on daily, weekly, monthly, quarterly or yearly basis. The following reports are available through PruneIt:
  - Job cost and profit report -This can be done for all jobs (specified time period) or for a specific job. This will assist company owners in evaluating their job costs and determining their accuracy
  - Time tracking report that shows estimated time to complete lawncare vs. actual time taken to complete lawncare
  - Expense report – This report will generate all the expenses for either a specific job or for the specified time period
  - Inactive customer report-list all customers that currently have no scheduled jobs. This report will allow the company to reach out to these customers such to maintain their business.

- Job report – Lists all the jobs within the specified time period.
- F5: Create Customer Invoices - The system will be able to automatically create invoices and bill customers after a job completion event has occurred. The invoice can be either printed or sent via email. The user may also use this feature to prepare an estimate in the event a customer wants one.
- F6: Input Job Schedules - This feature will enable users to input jobs and schedule them accordingly. This included the rescheduling of jobs as well. When inputting a job, users must also input the appropriate needs of the job such as landscapers needed, equipment, special requests, etc.
- F7: Input job types – users are able to input all the job types that the company offers and the cost for that job type. This will enable owners to select these job types when entering a new scheduled job for a customer.

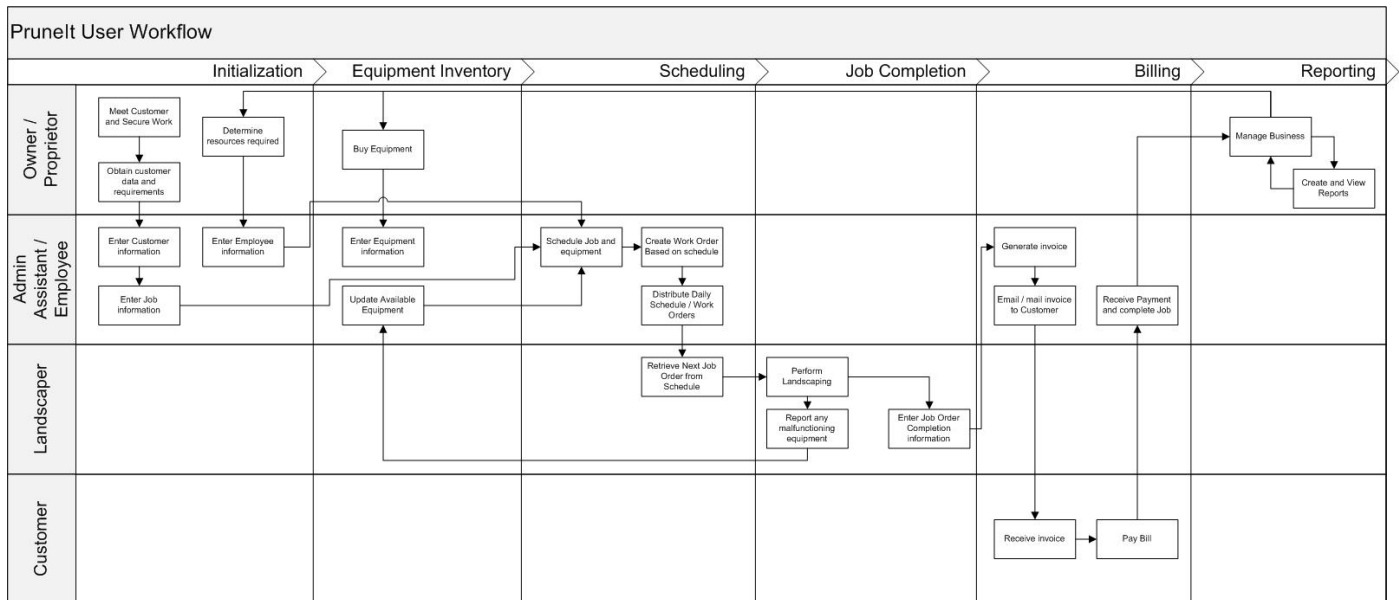
#### Landscapers and other employees (available via the mobile interface)

- F8: Track Landscaping Jobs- Landscapers will be able to input when a job starts and completes. The job will show as incomplete until it is logged as “started” or “complete”. Users will be able to see where each landscaper is or has been in addition to understanding the amount of work for each job (i.e. this job took this long and required this many landscapers). This will be offered as an on-screen tracking system that is online and updates in real time. There will also be the option for a printout with specifications of date and/or time.
- F9: Input Daily Work on Jobs by Employees - Users of this feature will be able to enter their time electronically. It will enable them to see how many hours they have worked in the current pay period and how many hours they are scheduled to work in the future.
- F10: View employee schedule – Users are able to view their job schedule for the day. This includes job information and special requests.
- F11: View job information – Users will be able to see all jobs and select a job to view the information and special requests
- F12: View company schedule – View all the job scheduled for the company
- F13: Input expenses – Users will be able to input expenses on the fly



### 3. System Use

#### 3.1 Support For User Workflows



#### 3.2 Actor Survey

##### Owner/Proprietor

Owner / Proprietor is responsible for identifying new clientele and selling landscaping services. This actor will interact with the system by creating and view reports. The use of these reports will allow this actor to track the business expenses, determine resource requirements.

Depending on the size of the organization, this actor may also be the Admin Assistant / Employee.

System Features:

- Generate / View reports.

##### Administrative Assistant/Employee

Administrative Assistant/Employee is the main user of the system. This actor is responsible for entering new clientele information from the salesman into the system. They are also responsible for maintaining clientele information. This actor is also responsible for entering employee information and Job information into the system. This actor will also develop the daily schedules for which equipment and landscapers are available for which jobs and create the work orders for the landscapers.

System Features:

- Enter / Maintain Clientele information
- Enter / Maintain employee information
- Enter / Maintain equipment information and availability.
- Enter / Maintain Job information.
- Create and distribute the Daily schedule of Job Orders.
- Generate Invoices
- Enter Payment information and complete jobs.

## **Landscaper**

Landscapers are the employees who are out there cutting the lawns and maintaining the clients gardens. They will receive their job orders for the day from the Administrative assistant. They will get/ check out the required equipment and go from client to client. If the landscaper has a smart phone, they will check-in at each job and also check out when the job is complete. If they do not have a smart phone, they will keep track of the time they start and stop each job and will enter the information into the new system when they return to the office. They will also supply the Administrative Assistant / Employee with the hours that they are available.

System Features:

- Enter job start time
- Enter job stop time
- Enter expenses
- View schedules
- View Job information

## Customers

The Customers are the ones receiving the services of the landscape company. They receive their service and then are invoiced for the work completed. They then pay for services rendered. At this time, the clients do not interact with the system.

## 3.2 Use-Case Model and System Events

### 3.2.1 Use-Case Model



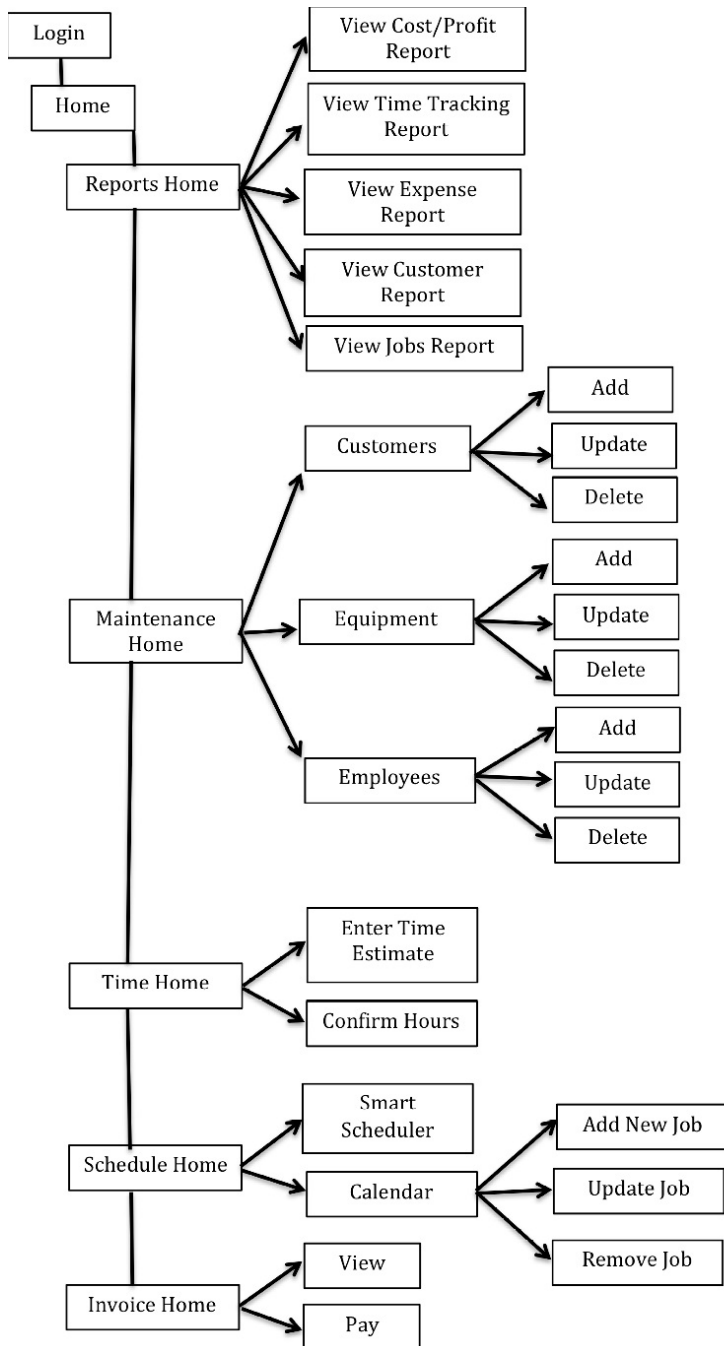
### 3.2.2 Event List

Business Event	System Action	System Interactions	Feature
New Customer	Enter Customer information	Enter customer information and job details	F1
New Employee	Enter Employee information	Enter Employee information and hours available to work	F2
New Equipment purchased	Enter Equipment information	Enter equipment information	F3
New Job is scheduled	Enter Job information	Enter job information	F7
Job is started	Job start date and time is updated	Landscaper enters start time and date via mobile app or web application	F8
Job is completed	Job completion date and time is updated, job time is calculated and stored	Landscaper enters completion time and date via mobile app or web application. They also enter all expenses.	F8, F13
Equipment Breaks down	Update equipment status	Equipment is taken out of service and is no longer available to schedule to a job	F3
Equipment is repaired	Update equipment status	Equipment is taken put back into service and is available to schedule to a job	F3
Daily Schedule is entered	System assigns landscaper and equipment to jobs and generates job orders	Employee enters and creates schedule information	F6

<b>Business Event</b>	<b>System Action</b>	<b>System Interactions</b>	<b>Feature</b>
	and prints job orders		
<b>Employee updates time worked</b>	Updates the employee hours worked	Employee enters information and can check hours worked	F9
<b>Employee checks work schedule</b>	Provide employee with work schedule	System retrieves schedule information for the logged in employee	F10
<b>Employee views Job information</b>	Provide employee with selected job status and information	System retrieves scheduled job information for the logged in employee	F11
<b>Employee checks work schedule for all Jobs</b>	Provide employee with work schedule for all jobs	System retrieves all scheduled job information and compiles for the logged in employee	F12
<b>Landscaper check hours worked</b>	Employees hours work are display	Select user requests a report for hours worked. Database is queried and results are displayed	F9
<b>Owner request reports</b>	Database queried for requested data and report is displayed	Select report	F4
<b>Job Completed</b>	Prepared invoice for email or printing	Compile job information and pricing for invoice	F5
<b>Payment received</b>	Enter payment	Complete invoice	F5

### 3.3 System Interfaces

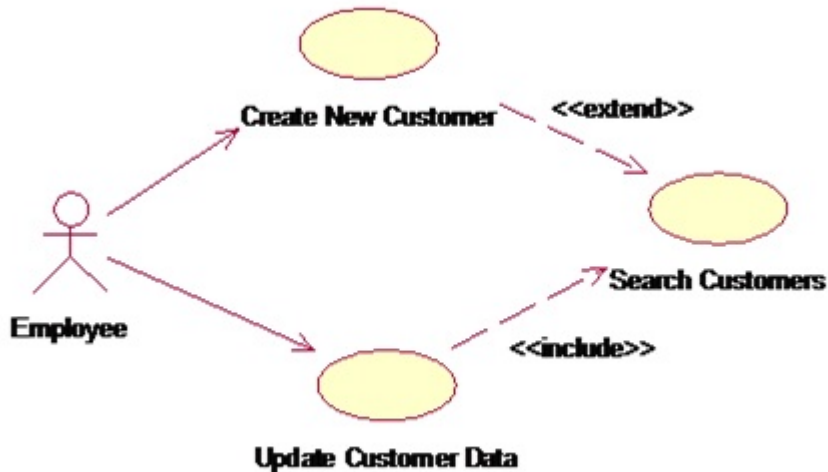
This section defines the main system interfaces to user features.



## 4. Specific requirements

### 4.1 System Use-Cases

#### Use Case 1: Input and Manage Customer Information



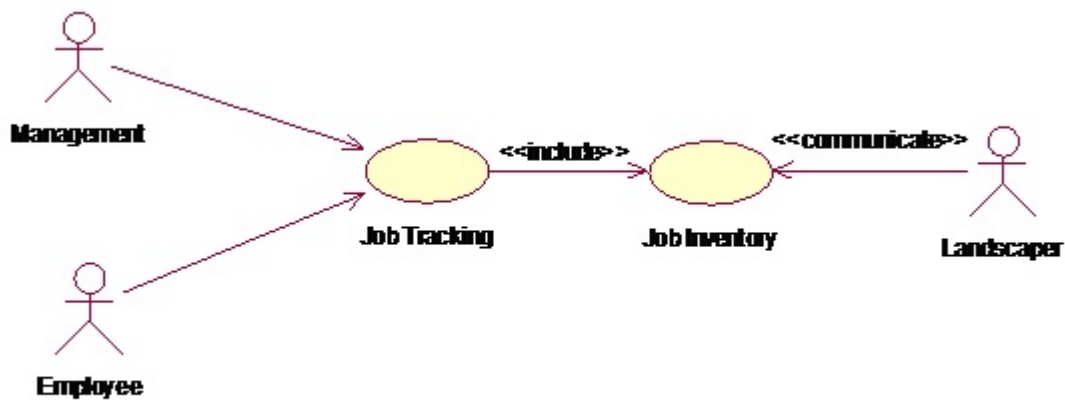
Use-case name	UC1 Input and Manage Customer Information
System or subsystem	PruneIt
Actors	Employee
Brief Description	This use case explains how an employee enters and stores all customer information such as name, address, telephone, email and customer's landscaping needs.
Basic flow of events	<p>Basic flow begins when a customer requests for landscaping services and is attended to by an employee. The employee selects Customer Information Management from the main menu:</p> <ol style="list-style-type: none"><li>1. The system displays a customer information entry screen to enter name, address, telephone, email and customer landscaping needs.</li><li>2. The employee enters the customer's information in the appropriate fields.</li><li>3. After details are entered, the system will display the customer record for the employee to verify that it is correct.</li></ol>

	<ol style="list-style-type: none"> <li>4. The system will then request the employee for the date the landscaping needs to be done.</li> <li>5. After the landscaping date is entered, the system will confirm that they have been entered.</li> <li>6. The system will then calculate the cost of the requested job and display it on the screen.</li> <li>7. The employee will then print the job summary and give it to the customer.</li> <li>8. The system will return to the main menu and then wait for further input from the employee.</li> <li>9. The employee will let the customer know that the company will confirm the job as soon as the scheduling is done.</li> </ol>
<i>Alternative flow of events</i>	<ol style="list-style-type: none"> <li>a. At any time, System Fails: To support recovery, ensure events can be recovered from any step of the scenario. <ol style="list-style-type: none"> <li>1. Employee restarts the system, logs in and requests recovery of prior state.</li> <li>2. System reconstructs prior state.</li> </ol> </li> <li>2a. System detects anomalies preventing recovery: <ol style="list-style-type: none"> <li>1. System signals error to the employee, records the error, and enters a clean state.</li> <li>2. Employee starts new customer information entry.</li> </ol> </li> <li>b. Printer out of paper. <ol style="list-style-type: none"> <li>1. If System can detect the fault, will signal the problem.</li> <li>2. Employee replaces paper.</li> <li>3. Employee requests another print out.</li> </ol> </li> </ol>
<i>Special requirements</i>	<p><u>Usability:</u> The system will present a data entry form that is intuitive to use for someone with high-school education. Employees should require minimal training to use the PruneIt system. The system will display waiting, failure, and input error messages in a clear and self-explanatory way.</p> <p><u>Performance:</u> The system should not take more than 5 seconds to store or locate and display customer information.</p> <p><u>Reliability:</u> The system will create a restore-point after each customer input. If</p>



	an error is encountered, the system will roll back to the previous restore-point, then return control to the user.
<i>Pre-conditions</i>	The employee must be identified, authenticated, logged into the system and at the main menu screen.
<i>Post-condition(s)</i>	The employee has system confirmation that customer information has been stored or was not completed and an error message shown to the user and the system awaits further user input. PruneIt's database must contain the entered customer information. Confirmation of job request printed by the system.
<i>Extension points</i>	1. Search Customers

## Use Case 2: Tracking Landscaping Jobs

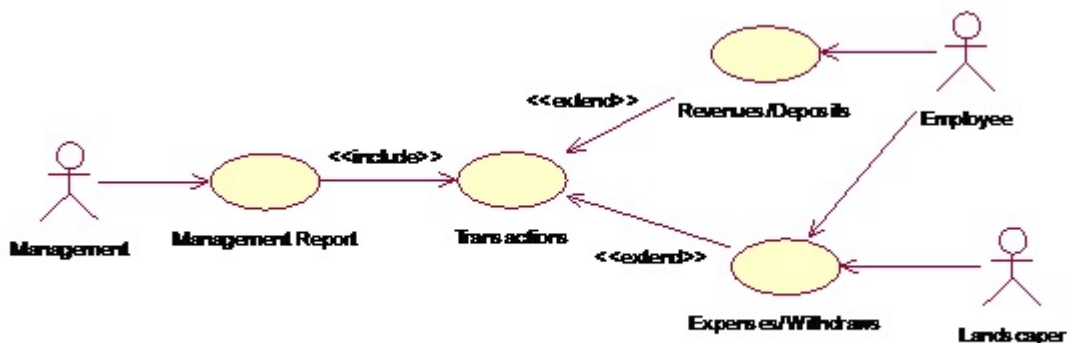


<i>Use-case name</i>	UC2 Tracking Landscaping Jobs
<i>System or subsystem</i>	PruneIt
<i>Actors</i>	Management, Employee
<i>Brief Description</i>	This use case explains how the manager or employee tracks where each landscaper is or has been in addition to understanding the amount of work for each job (i.e. this job took this long and required this many landscapers). This will be offered as an on-screen tracking system that is online and updates in real time. There will also be the option for a printout with specifications of date and/or time.

<p><i>Basic flow of events</i></p>	<p>Basic flow begins when a manager or employee wants to check on jobs being done and those completed. The manager or employee selects Job Tracking from the main menu:</p> <ol style="list-style-type: none"> <li>1. The system displays a tracking period entry screen that has a default of the current day according to the system clock. The user has a choice of current day, week, month, year or defined period from a pull down menu.</li> <li>2. The user selects the tracking period they want.</li> <li>3. The systems displays a landscaper selection screen that has a default of the all landscapers currently assigned jobs. The user has a choice of all landscapers or can choose a specific landscaper or select the landscapers they want to track.</li> <li>4. The user enters the landscaper selection criteria they want.</li> <li>5. The system displays the jobs according to the period specified and the landscapers selected. The information is displayed landscaper, job number, current status (done/being done/awaiting), schedule status (online/late).</li> <li>6. The user can then print the job tracking summary.</li> <li>7. The system will return to the main menu and then wait for further input from the user.</li> </ol>
<p><i>Alternative flow of events</i></p>	<ol style="list-style-type: none"> <li>a. At any time, System Fails: To support recovery, ensure events can be recovered from any step of the scenario. <ol style="list-style-type: none"> <li>1. Manager/Employee restarts the system, logs in and requests recovery of prior state.</li> <li>2. System reconstructs prior state. <ol style="list-style-type: none"> <li>2a. System detects anomalies preventing recovery: <ol style="list-style-type: none"> <li>1. System signals error to the manager/employee, records the error, and enters a clean state.</li> <li>2. Manager/Employee starts new job tracking session.</li> </ol> </li> </ol> </li> </ol> </li> <li>b. Printer out of paper. <ol style="list-style-type: none"> <li>1. If System can detect the fault, will signal the problem.</li> </ol> </li> </ol>

	2. Manager /Employee replaces paper. 3. Manager /Employee requests another print out.
<i>Special requirements</i>	<p><u>Usability:</u>          The system will present a job tracking screen that is intuitive to use for someone with high-school education. Manager /Employees should require minimal training to use the PruneIt system.</p> <p>The system will display waiting, failure, and input error messages in a clear and self-explanatory way.</p> <p><u>Performance:</u>          The system should not take more than 5 seconds to locate and display job tracking information.</p> <p><u>Reliability:</u> If an error is encountered, the system will return control to the user.</p>
<i>Pre-conditions</i>	The Manager /employee must be identified, authenticated, logged into the system and at the main menu screen.
<i>Post-condition(s)</i>	The Manager /employee has system confirmation on the jobs being tracked. Confirmation of job tracking information printed by the system.
<i>Extension points</i>	None

### Use Case 3: Create Management Report

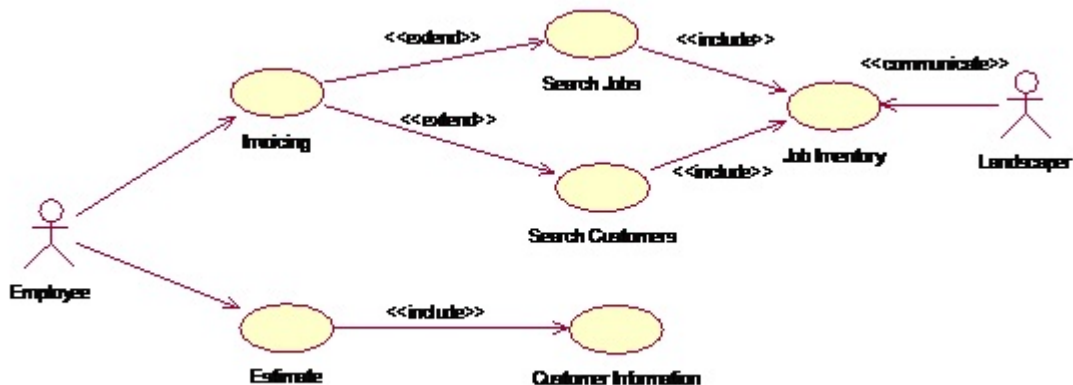


<i>Use-case name</i>	UC3 Create Management Report
----------------------	------------------------------

<i>System or subsystem</i>	PruneIt
<i>Actors</i>	Management
<i>Brief Description</i>	This use case explains how a manager creates a report on profitability that shows how much revenue is coming in against the expenses being incurred. Report can be set for period of time e.g. daily, weekly, monthly, and yearly or any range of dates.
<i>Basic flow of events</i>	<p>Basic flow begins when a manager wants to monitor revenue against expenses that are being recorded. The manager selects Management Report from the main menu:</p> <ol style="list-style-type: none"> <li>1. The system displays a reporting period entry screen that has a default of the current week according to the system clock. The user has a choice of current week, month, year or defined period from a pull down menu.</li> <li>2. The user selects the reporting period they want.</li> <li>3. The system displays the revenue vs. expenses report according to the period specified. The information is displayed Revenue, Expenses, with totals at the end and profitability.</li> <li>4. The user can then print the management report.</li> <li>5. The system will return to the main menu and then wait for further input from the employee.</li> </ol>
<i>Alternative flow of events</i>	<p>a. At any time, System Fails: To support recovery, ensure events can be recovered from any step of the scenario.</p> <ol style="list-style-type: none"> <li>1. Manager restarts the system, logs in and requests recovery of prior state.</li> <li>2. System reconstructs prior state.</li> </ol> <p>2a. System detects anomalies preventing recovery:</p> <ol style="list-style-type: none"> <li>1. System signals error to the manager, records the error, and enters a clean state.</li> <li>2. Manager starts new job tracking session.</li> </ol> <p>b. Printer out of paper.</p> <ol style="list-style-type: none"> <li>1. If System can detect the fault, will signal the problem.</li> <li>2. Manager replaces paper.</li> <li>3. Manager requests another print out.</li> </ol>

<i>Special requirements</i>	<p><u>Usability:</u> The system will present a management report screen that is intuitive to use for someone with high-school education. Manager should require minimal training to use the PruneIt system.</p> <p>The system will display waiting, failure, and input error messages in a clear and self-explanatory way.</p> <p><u>Performance:</u> The system should not take more than 5 seconds to locate and display management report information.</p> <p><u>Reliability:</u> If an error is encountered, the system will return control to the user.</p>
<i>Pre-conditions</i>	The manager must be identified, authenticated, logged into the system and at the main menu screen.
<i>Post-condition(s)</i>	<p>The Manager has system confirmation on the revenues and expenses being monitored.</p> <p>Confirmation of management report printed by the system.</p>
<i>Extension points</i>	

## Use Case 4: Create Customer Invoices



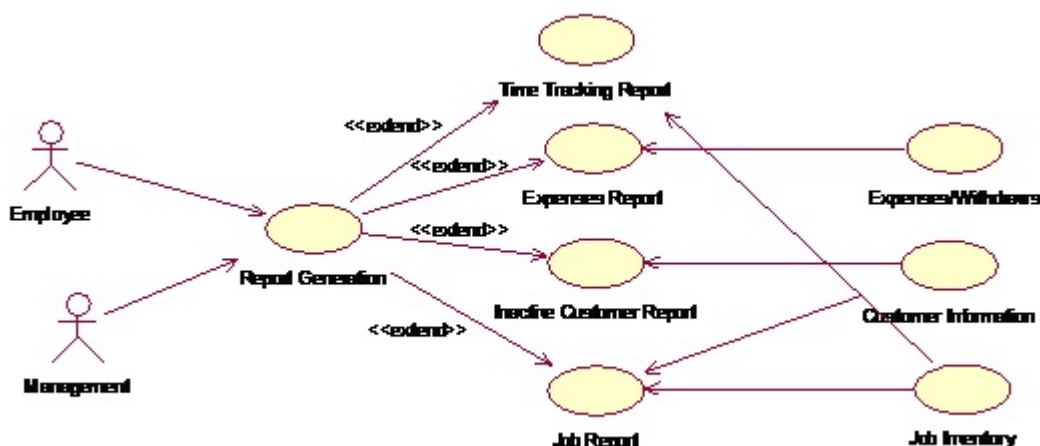
<i>Use-case name</i>	UC4 Create Customer Invoices
<i>System or subsystem</i>	PruneIt

<i>Actors</i>	Employee
<i>Brief Description</i>	This use case explains how an employee creates invoices and bill customers after a job completion event has occurred. The invoice can be either printed or sent via email. The user may also use this feature to prepare an estimate in the event a customer wants one.
<i>Basic flow of events</i>	<p>Basic flow begins when an employee wants to create an invoice or prepare an estimate for a customer. The employee selects Customer Invoicing from the main menu and selects Create Invoice from the sub-menu:</p> <ol style="list-style-type: none"> <li>1. The system presents the employee with the Invoicing interface.</li> <li>2. The employee searches by customer or job and executes the search.</li> <li>3. The system returns the results.</li> <li>4. The employee chooses a job for which to create an invoice.</li> <li>5. The system calculates the total amount due based on job details and creates the invoice.</li> <li>6. The employee may then emails, or prints the invoice.</li> <li>7. The system will return to the main menu and then wait for further input from the employee.</li> </ol>
<i>Alternative flow of events</i>	<p>a. At any time, System Fails:</p> <p>To support recovery, ensure events can be recovered from any step of the scenario.</p> <ol style="list-style-type: none"> <li>1. Employee restarts the system, logs in and requests recovery of prior state.</li> <li>2. System reconstructs prior state.</li> </ol> <p>2a. System detects anomalies preventing recovery:</p> <ol style="list-style-type: none"> <li>1. System signals error to the employee, records the error, and enters a clean state.</li> <li>2. employee starts new customer invoicing session.</li> </ol> <p>b. Printer out of paper.</p> <ol style="list-style-type: none"> <li>1. If System can detect the fault, will signal the problem.</li> <li>2. Employee replaces paper.</li> <li>3. Employee requests another print out.</li> </ol>

	<p>c. The Employee wants to create an estimate for the customer. The employee selects Customer Invoicing from the main menu and selects Create Estimate from the sub-menu:</p> <ol style="list-style-type: none"> <li>1. The system displays a customer information entry screen to enter name, address, telephone, email and customer landscaping needs.</li> <li>2. The employee enters the customer's information in the appropriate fields.</li> <li>3. After details are entered, the system will display the customer record for the employee to verify that it is correct.</li> <li>4. The system will then calculate the cost of the requested job and display it on the screen.</li> <li>5. The employee will then print the job estimate and give it to the customer.</li> <li>6. The system will return to the main menu and then wait for further input from the employee.</li> </ol>
<i>Special requirements</i>	<p><u>Security</u>: Only the Invoicing/ Accounts Receivables Administrator may execute Customer Invoicing.</p> <p><u>Usability</u>: The system will present a data entry form that is intuitive to use for someone with high-school education. Employees should require minimal training to use the PruneIt system. The system will display waiting, failure, and input error messages in a clear and self-explanatory way.</p> <p><u>Performance</u>: The system should not take more than 5 seconds to store or locate and display customer information.</p> <p><u>Reliability</u>: The system will create a restore-point after each customer input. If an error is encountered, the system will roll back to the previous restore-point, then return control to the user.</p>
<i>Pre-conditions</i>	The employee must be identified, authenticated, logged into the system and at the main menu screen.
<i>Post-condition(s)</i>	<p>The employee has system confirmation that customer information has been stored or was not completed and an error message shown to the user and the system awaits further user input.</p> <p>PruneIt's database must contain the entered customer information.</p> <p>Confirmation of job estimate printed by the system.</p>

<i>Extension points</i>	1. Search Jobs 2. Search Customers
-------------------------	---------------------------------------

## Use Case 5: Create Reports



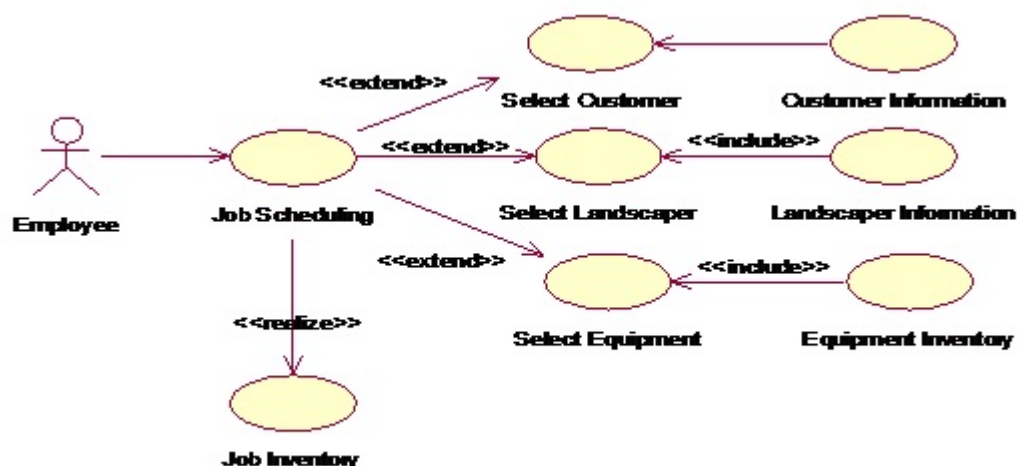
<i>Use-case name</i>	UC5 Create Reports
<i>System or subsystem</i>	PruneIt
<i>Actors</i>	Employee, Management
<i>Brief Description</i>	This use case explains how an employee or manager will have the option to create reports on daily, weekly, monthly, quarterly or yearly basis. Available reports will be Time tracking report, Expense report, Inactive Customer Report and Job Report.
<i>Basic flow of events</i>	<p>Basic flow begins when an employee or manager wants to create a report.  The user selects Reports from the main menu:</p> <ol style="list-style-type: none"> <li>1. The system displays a pull down submenu for the employee to select a report to create</li> <li>2. User selects the report to create.</li> <li>3. The system prompts the user for the reporting period.</li> <li>4. The user specifies the reporting period.</li> <li>5. The system generates the desired report.</li> <li>6. The user can either view the report on screen or print.</li> </ol>



	7. The system will return to the main menu and then wait for further input from the user.
<i>Alternative flow of events</i>	<p>a. At any time, System Fails: To support recovery, ensure events can be recovered from any step of the scenario.</p> <ol style="list-style-type: none"> <li>1. Employee or Manager restarts the system, logs in and requests recovery of prior state.</li> <li>2. System reconstructs prior state.</li> </ol> <p>2a. System detects anomalies preventing recovery:</p> <ol style="list-style-type: none"> <li>1. System signals error to the employee or manager, records the error, and enters a clean state.</li> <li>2. Employee/Manager starts new job tracking session.</li> </ol> <p>b. Printer out of paper.</p> <ol style="list-style-type: none"> <li>1. If System can detect the fault, will signal the problem.</li> <li>2. Employee/Manager replaces paper.</li> <li>3. Employee/Manager requests another print out.</li> </ol>
<i>Special requirements</i>	<p><u>Usability:</u> The system will present a data entry form that is intuitive to use for someone with high-school education. Employees/Manager should require minimal training to use the PruneIt system. The system will display waiting, failure, and input error messages in a clear and self-explanatory way.</p> <p><u>Performance:</u> The system should not take more than 5 seconds to store or locate and display customer information.</p> <p><u>Reliability:</u> The system will create a restore-point after each customer input. If an error is encountered, the system will roll back to the previous restore-point, then return control to the user.</p>
<i>Pre-conditions</i>	The employee/manager must be identified, authenticated, logged into the system and at the main menu screen.
<i>Post-condition(s)</i>	The employee/manager has system confirmation on data being monitored. Confirmation of reports printed by the system.
<i>Extension points</i>	<ol style="list-style-type: none"> <li>1. Time tracking report</li> <li>2. Expense report</li> </ol>

	3. Inactive Customer Report 4. Job Report
--	--

## Use Case 6: Input Job Schedules

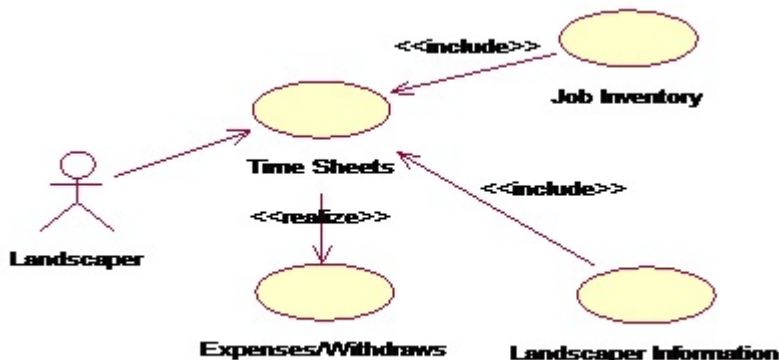


Use-case name	UC6 Input Job Schedules
System or subsystem	PruneIt
Actors	Employee
Brief Description	This use case explains how an employee will input jobs and schedule them accordingly. This includes the rescheduling of jobs as well. When inputting a job, users must also input the appropriate needs of the job such as landscapers needed, equipment, special requests, etc.
Basic flow of events	<p>Basic flow begins when an employee wants to schedule jobs. The employee selects Job Schedule from the main menu:</p> <ol style="list-style-type: none"> <li>1. The system requests the user to enter for which day to schedule the jobs.</li> <li>2. The employee enters the day from a pull down menu of days for current month.</li> <li>3. The system displays a job scheduling screen with three areas – unattended customers, unassigned landscapers and unassigned equipment.</li> <li>4. The employee selects the customer to schedule a job</li> </ol>

	<p>for.</p> <ol style="list-style-type: none"> <li>5. The employee selects the landscaper(s) assigned to the job.</li> <li>6. The employee selects the equipment for the job.</li> <li>7. The employee clicks “add” to job schedule.</li> <li>8. The system adds the job to schedule for the selected day.</li> <li>9. The system requests the user to enter another job or exit to main menu.</li> <li>10. The user can either view the schedule on screen or print.</li> <li>11. The system will return to the main menu and then wait for further input from the user.</li> </ol>
<i>Alternative flow of events</i>	<ol style="list-style-type: none"> <li>a. At any time, System Fails: To support recovery, ensure events can be recovered from any step of the scenario. <ol style="list-style-type: none"> <li>1. Employee restarts the system, logs in and requests recovery of prior state.</li> <li>2. System reconstructs prior state.</li> </ol> </li> <li>2a. System detects anomalies preventing recovery: <ol style="list-style-type: none"> <li>1. System signals error to the employee, records the error, and enters a clean state.</li> <li>2. Employee starts new input job schedule session.</li> </ol> </li> <li>b. Printer out of paper. <ol style="list-style-type: none"> <li>1. If System can detect the fault, will signal the problem.</li> <li>2. Employee replaces paper.</li> <li>3. Employee requests another print out.</li> </ol> </li> </ol>
<i>Special requirements</i>	<p><u>Usability:</u> The system will present a data select form that is intuitive to use for someone with high-school education. Employees should require minimal training to use the PruneIt system. The system will display waiting, failure, and input error messages in a clear and self-explanatory way.</p> <p><u>Performance:</u> The system should not take more than 5 seconds to locate and display customer, landscape and equipment information.</p>

	<u>Reliability</u> : The system will create a restore-point after each job input. If an error is encountered, the system will roll back to the previous restore-point, then return control to the user.
<i>Pre-conditions</i>	The employee must be identified, authenticated, logged into the system and at the main menu screen.
<i>Post-condition(s)</i>	The employee has system confirmation that job schedule has been stored or was not completed and an error message shown to the user and the system awaits further user input. PruneIt's database must contain the created job schedule. Confirmation of job schedule printed by the system.
<i>Extension points</i>	1. Select Customer 2. Select Landscaper 3. Select Equipment 4. Job Inventory

## Use Case 7: Input Daily Work On Jobs By Landscapers

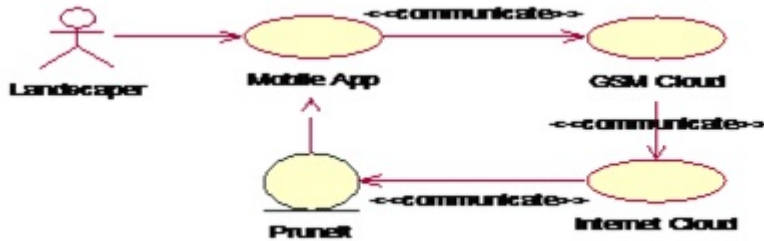


<i>Use-case name</i>	UC7 Input Daily Work On Jobs By Landscaper
<i>System or subsystem</i>	PruneIt
<i>Actors</i>	Landscaper
<i>Brief Description</i>	This use case explains how landscapers will enter their daily job time electronically. It will enable them to see how many hours they have worked in the current pay period and how many hours they are scheduled

	to work in the future.
<i>Basic flow of events</i>	<p>Basic flow begins when a landscaper wants to enter daily job hours for landscaping jobs done. The landscaper selects Daily Job Hours from the main menu and selects Input Hours from the sub-menu:</p> <ol style="list-style-type: none"> <li>1. The system displays a daily job hours input entry screen with name already pre-filled.</li> <li>2. The landscaper selects the day for hours to enter from a day field combobox of days for the month.</li> <li>3. The landscaper enters the hours worked in the hours field.</li> <li>4. The landscaper selects from a job field combobox pre-filled with jobs for that day assigned to the landscaper.</li> <li>5. The landscaper clicks “add”.</li> <li>6. The system confirms the information entered by the Landscaper.</li> <li>7. The system will request the landscaper if they want to enter more hours.</li> <li>8. The landscaper can enter more hours or go to the main menu.</li> <li>9. The system will then calculate total hours worked for current pay period and the pay payable and display on the screen.</li> <li>10. The landscaper will then print the daily job hours summary.</li> <li>11. The system will return to the main menu and then wait for further input from the landscaper.</li> </ol>
<i>Alternative flow of events</i>	<p>a. At any time, System Fails: To support recovery, ensure events can be recovered from any step of the scenario.</p> <ol style="list-style-type: none"> <li>1. Employee restarts the system, logs in and requests recovery of prior state.</li> <li>2. System reconstructs prior state.</li> </ol> <p>2a. System detects anomalies preventing recovery:</p> <ol style="list-style-type: none"> <li>1. System signals error to the employee, records the error, and enters a clean state.</li> <li>2. Landscaper logs in.</li> </ol>

	<p>3. Landscaper starts new input daily work on job session.</p> <p>b. Printer out of paper.</p> <ol style="list-style-type: none"> <li>1. If System can detect the fault, will signal the problem.</li> <li>2. Landscaper/employee replaces paper.</li> <li>3. Landscaper requests another print out.</li> </ol> <p>c. The landscaper wants to see hours scheduled for the future.</p> <ol style="list-style-type: none"> <li>1. The landscaper selects Daily Job Hours from the main menu:</li> <li>2. The Landscaper selects Future Hours from the submenu</li> <li>3. The system generates the future hours to work and displays on the screen.</li> <li>4. The Landscaper can then print the future hours to work.</li> <li>5. The system will return to the main menu and then wait for further input from the landscaper.</li> </ol>
<i>Special requirements</i>	<p><u>Usability:</u> The system will present a daily job hour entry form that is intuitive to use for someone with high-school education. Landscapers should require minimal training to use the PruneIt system via the web.</p> <p>The system will display waiting, failure, and input error messages in a clear and self-explanatory way.</p> <p><u>Performance:</u> The system should not take more than 5 seconds to locate and display information.</p> <p><u>Reliability:</u> The system will create a restore-point after each daily job hour input. If an error is encountered, the system will roll back to the previous restore-point, then return control to the user.</p>
<i>Pre-conditions</i>	The landscaper must be identified, authenticated, logged into the system and at the main menu screen.
<i>Post-condition(s)</i>	<p>The landscaper has system confirmation that daily job hours have been stored or was not completed and an error message shown to the user and the system awaits further user input.</p> <p>PruneIt's database must contain the input daily job hours.</p> <p>Confirmation of daily job hours printed by the system.</p>
<i>Extension points</i>	1. Expenses/Withdrawals

## Use Case 8: Access PruneIT Information From Mobile Devices



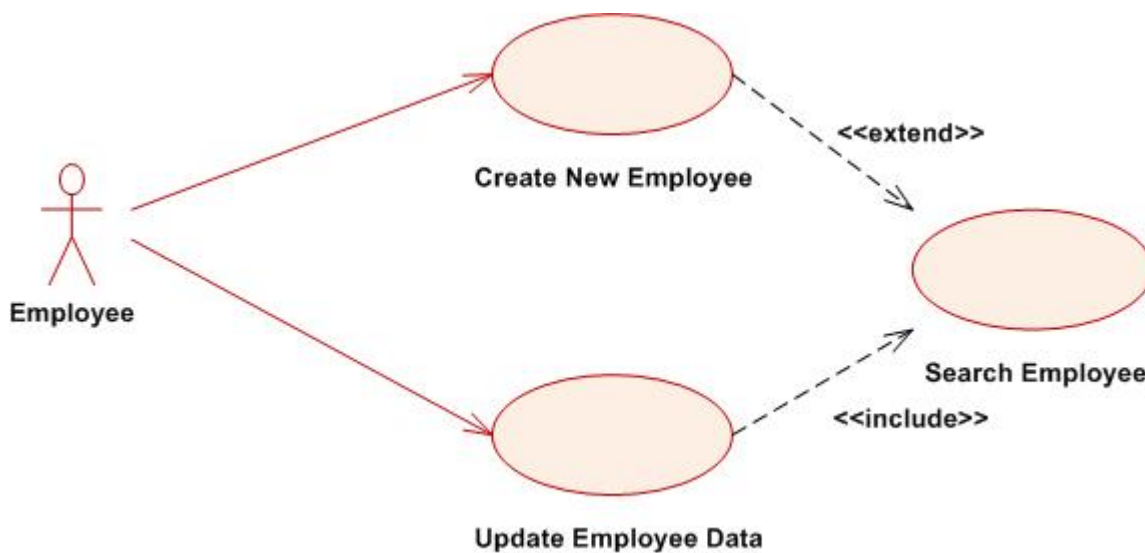
<i>Use-case name</i>	UC8 Access PruneIT Information From Mobile Devices
<i>System or subsystem</i>	Mobile Application
<i>Actors</i>	Landscaper
<i>Brief Description</i>	This use case explains how a landscaper can access important information such as scheduling and contact information in the field from a mobile device. The system will be able to handle rudimentary system tasks, such as job completion and report generation. The following features will be available via the mobile device:
<i>Basic flow of events</i>	<p>Basic flow begins when a landscaper wants to access information from a mobile device:</p> <ol style="list-style-type: none"> <li>1. The landscaper starts the mobile application on the mobile device.</li> <li>2. The mobile application requests log in from the PruneIt server.</li> <li>3. PruneIt server provides log in access.</li> <li>4. The landscaper logs into the PruneIt server.</li> <li>5. PruneIt server identifies, authenticates and logs in the landscaper.</li> <li>6. The Landscaper logs on to PruneIt.</li> <li>7. The PruneIt main menu is enabled through the mobile application.</li> <li>8. The landscape can access the various menu options that he has permissions for.</li> </ol>

<i>Alternative flow of events</i>	<p>a. At any time, System Fails:</p> <p>To support recovery, ensure events can be recovered from any step of the scenario.</p> <ol style="list-style-type: none"> <li>1. Employee restarts the system, logs in and requests recovery of prior state.</li> <li>2. System reconstructs prior state.</li> </ol> <p>2a. System detects anomalies preventing recovery:</p> <ol style="list-style-type: none"> <li>1. System signals error to the employee and mobile app, records the error, and enters a clean state.</li> <li>2. Landscaper logs in.</li> <li>3. Landscaper starts new PruneIt session.</li> </ol> <p>b. At any time, GSM system fails:</p> <p>GSM failure cannot support recovery. The landscaper must initiate the following steps upon recovery of GSM to normal operating state:</p> <ol style="list-style-type: none"> <li>1. Start Mobile Application.</li> <li>2. Log on to PruneIt server.</li> <li>3. Log on to PruneIt web application.</li> </ol>
<i>Special requirements</i>	<p><u>Security</u>: Mobile application requires user to authenticated by PruneIt server.</p> <p><u>Usability</u>:</p> <p>The system will present the PruneIt main menu that is intuitive to use for someone with high-school education. Landscapers should require minimal training to use the PruneIt system.</p> <p>The system will display waiting, failure, and input error messages in a clear and self-explanatory way.</p> <p><u>Performance</u>:</p> <p>The system should not take more than 5 seconds to locate and display customer, landscape and equipment information. Performance is highly dependent on Internet performance.</p> <p><u>Reliability</u>: The system will create a restore-point after each job input. If an error is encountered, the system will roll back to the previous restore-point, then return control to the user. However, internet connectivity reliability is a major factor.</p>
<i>Pre-conditions</i>	<ol style="list-style-type: none"> <li>1. Landscaper must have access to a running Android application with internet access via either WiFi or Mobile Broadband.</li> <li>2. The landscaper must be identified, authenticated, logged into the system and at the main menu screen.</li> </ol>
<i>Post-condition(s)</i>	The landscaper has access to PruneIt information or was not completed and



	an error message shown to the user and the system awaits further user input.
<i>Extension points</i>	

## Use Case 9: Input and Manage Employee Information

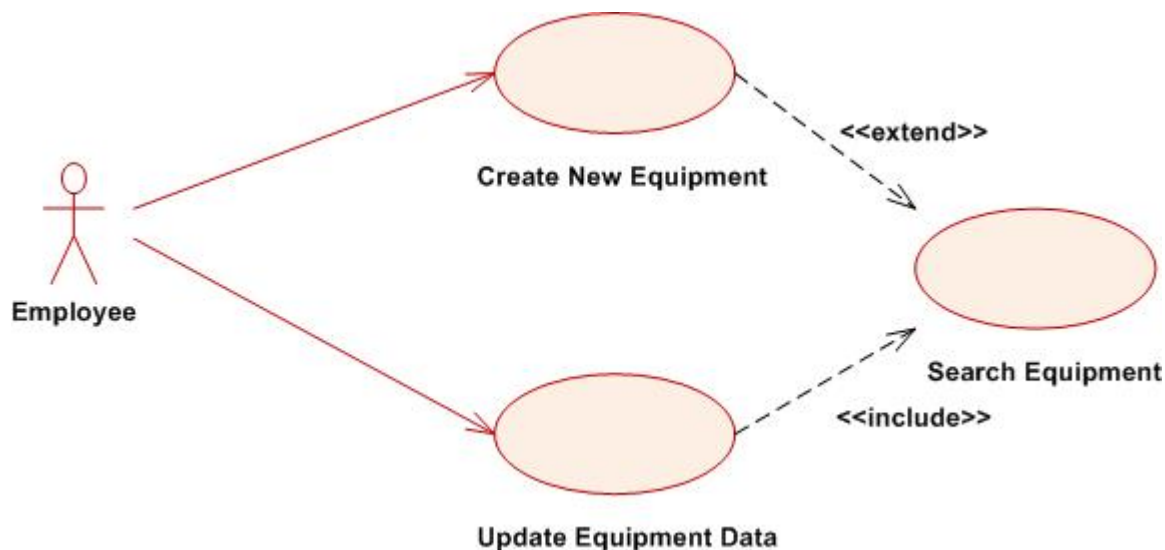


<i>Use-case name</i>	<b>UC9 Input and Manage Employee Information</b>
<i>System or subsystem</i>	PruneIt
<i>Actors</i>	Employee
<i>Brief Description</i>	This use case explains how an employee enters and stores all new employee information such as name, address, telephone, email and the employees scheduling needs.
<i>Basic flow of events</i>	<p>Basic flow begins when a new employee is hired or moves. The employee selects Employee Information Management from the main menu:</p> <ol style="list-style-type: none"> <li>1. The system displays a employee information entry screen to enter name, address, telephone, email and scheduling needs.</li> <li>2. The employee enters the employee's information into</li> </ol>

	<p>the appropriate fields.</p> <p>3. After details are entered, the system will display the employee record for the employee to verify that it is correct.</p>
<i>Alternative flow of events</i>	<p>a. At any time, System Fails:</p> <p>To support recovery, ensure events can be recovered from any step of the scenario.</p> <ol style="list-style-type: none"> <li>1. Employee restarts the system, logs in and requests recovery of prior state.</li> <li>2. System reconstructs prior state. <ol style="list-style-type: none"> <li>2a. System detects anomalies preventing recovery: <ol style="list-style-type: none"> <li>1. System signals error to the employee, records the error, and enters a clean state.</li> <li>2. Employee starts new employee information entry.</li> </ol> </li> </ol> </li> </ol> <p>b. Printer out of paper.</p> <ol style="list-style-type: none"> <li>1. If System can detect the fault, will signal the problem.</li> <li>2. Employee replaces paper.</li> <li>3. Employee requests another print out.</li> </ol>
<i>Special requirements</i>	<p><u>Usability:</u></p> <p>The system will present a data entry form that is intuitive to use for someone with high-school education. Employees should require minimal training to use the PruneIt system.</p> <p>The system will display waiting, failure, and input error messages in a clear and self-explanatory way.</p> <p><u>Performance:</u></p> <p>The system should not take more than 5 seconds to store or locate and display employee information.</p> <p><u>Reliability:</u> The system will create a restore-point after each Employee input. If an error is encountered, the system will roll back to the previous restore-point, and then return control to the user.</p>

<i>Pre-conditions</i>	The employee must be identified, authenticated, logged into the system and at the main menu screen.
<i>Post-condition(s)</i>	The employee has system confirmation that entered employee's information has been stored or was not completed and an error message shown to the user and the system awaits further user input.  PruneIt's database must contain the entered employee information.
<i>Extension points</i>	1. Search Employees

### Use Case 10: Input and Manage Equipment Information



<i>Use-case name</i>	<b>UC10 Input and Manage Equipment Information</b>
<i>System or subsystem</i>	PruneIt
<i>Actors</i>	Employee
<i>Brief Description</i>	This use case explains how an employee enters and stores all new equipment information such as make, model, status and the equipment scheduling needs.
<i>Basic flow of events</i>	Basic flow begins when a new piece of equipment is purchased or requires updating. The employee selects Equipment Information Management from the

	<p>main menu:</p> <ol style="list-style-type: none"> <li>1. The system displays a Equipment information entry screen to enter make, model, status and the equipment scheduling needs.</li> <li>2. The employee enters the equipment's information into the appropriate fields.</li> <li>3. After details are entered, the system will display the equipment record for the employee to verify that it is correct.</li> </ol>
<i>Alternative flow of events</i>	<ol style="list-style-type: none"> <li>a. At any time, System Fails: <p>To support recovery, ensure events can be recovered from any step of the scenario.</p> <ol style="list-style-type: none"> <li>1. Employee restarts the system, logs in and requests recovery of prior state.</li> <li>2. System reconstructs prior state.</li> </ol> </li> <li>2a. System detects anomalies preventing recovery: <ol style="list-style-type: none"> <li>1. System signals error to the employee, records the error, and enters a clean state.</li> <li>2. Employee starts new employee information entry.</li> </ol> </li> <li>b. Printer out of paper. <ol style="list-style-type: none"> <li>1. If System can detect the fault, will signal the problem.</li> <li>2. Employee replaces paper.</li> <li>3. Employee requests another print out.</li> </ol> </li> </ol>
<i>Special requirements</i>	<p><u>Usability:</u></p> <p>The system will present a data entry form that is intuitive to use for someone with high-school education. Employees should require minimal training to use the PruneIt system.</p> <p>The system will display waiting, failure, and input error messages in a clear and self-explanatory way.</p> <p><u>Performance:</u></p> <p>The system should not take more than 5 seconds to store or locate and display</p>

	<p>employee information.</p> <p><u>Reliability</u>: The system will create a restore-point after each Employee input. If an error is encountered, the system will roll back to the previous restore-point, and then return control to the user.</p>
<i>Pre-conditions</i>	The employee must be identified, authenticated, logged into the system and at the main menu screen.
<i>Post-condition(s)</i>	<p>The employee has system confirmation that entered equipment's information has been stored or was not completed and an error message shown to the user and the system awaits further user input.</p> <p>PruneIt's database must contain the entered equipment information.</p>
<i>Extension points</i>	1. Search Equipment

## 4.2 System Functional Specification

This section provides an overview of the high-level functional processes required for PruneIt. These are categorized by software product component (See Figure 2.2).

### Client Terminals – User Interface Functions

As PruneIt is a web application, the system architecture employed will be client-server. All applications will normally execute on client hardware with data requests facilitated through the server. However, mobile application accessibility will be the only exception, which will request server hardware functionality. Each of user interface function will be provided by one of the server modules. This module displays user screens and generates HTTP requests to trigger server application processing.

*In Figure 3.2, the following functions are fully populated:*

CT1: User login and authorization; main menu screen.

CT2: Main menu screen (options dependent on login access rights).

CT3: Generate and view / print reports (options dependent on login access rights).

CT4: Enter or maintain customer information (manual data entry at terminal).

CT5: Enter or maintain employee information (manual data entry at terminal).

CT6: Enter or maintain job information (manual data entry at terminal).

CT7: Enter or maintain equipment inventory and availability information (manual data entry at terminal).

CT8: Enter job start time (manual data entry at terminal).

CT9: Enter job completion time (manual data entry at terminal).

CT10: Enter available work hours (manual data entry at terminal).

CT11: View schedule (manual data entry at terminal).

*The following function is only partly populated, as the full configuration options will be determined in Version 2.0 of this specification (not covered by this specification):*

CT12: System management and configuration (Manual data entry terminal)

*The following functions are not shown in Figure 3.2, as these will be provided as automatic functions to the Customer Invoicing interface:*

CT13: Tax calculation (automated)

CT14: Overdue charge (automated)

CT15: Bounced check or unauthorized credit/debit card fee (automated)

## **Server Application Modules**

Each of the high-level functions listed below will be supported by a server application module and will be accessed via the client interface, with the exception of System Management and Configuration functions, which will be accessible by direct access to the server only. System functions have been grouped by application module.

### **System Management and Configuration**

These functions will be limited to direct access from server, for system security protection.

SYS1: Configure system parameters (administrator only).

SYS2: Create and configure user accounts and access-rights.

SYS3: Configure management report parameters.

SYS4: Configure module access permissions.

SYS5: Configure mobile application access.

SYS6: View or print system use audit reports.

### **Job Scheduling**

JBS1: Enables the user select customer to schedule a job for.

JBS2: Enables the user select landscapers for a job.

JBS3: Enables the user select equipment for a job.

JBS4: Provides for automatic insertion of jobs into the job inventory.

JBS5: Enables the user update / change job requirements.

JBS6: Enables the user cancel or reschedule jobs.

JBS7: Enables the user confirm completion of a job.

### **Customer Storage**

CTS1: Enables the creation of customer records.

CTS2: Enables the update of customer records.

CTS3: Enables the deletion of customer records.

### **Employee Storage**

EMS1: Enables the creation of employee records.

EMS2: Enables the update of employee records.

EMS3: Enables the deletion of employee records.

### **Report Generation**

RTG1: Enables users to view or print time tracking report that shows hours worked by each landscaper over a specified period.

RTG2: Enables users to view or print expense report that shows expenses incurred over a specified period and categorized by type of expense.

RTG3: Enables users to view or print inactive customer report that shows which customers have not had any landscaping jobs done for a specified period.

RTG4: Enables users to view or print job report that shows jobs done over a specified period.

RTG5: Enables user to view or print management report that shows revenues against expenses (Profitability). This report is strictly by management access only.

### **Job Types**

JBT1: Provides access to job type addition and deletion.

JBT2: Enables users to change job type details and pricing.

### **Time Tracking**

TTK1: Enables the user enter daily hours worked.

TTK2: Enables the user view total hours worked in current pay period.

TTK3: Enables the user view hours to work in the future.

### **Job Tracking**

JTK1: Enables the location of each landscaper.

JTK2: Enables the user view the amount of work for the job.

JTK3: Enables the user view the amount of time for the job.

JTK4: Enables the user view how much labor is assigned for the job.

### **Billing**

BIL1: Provides for the automatic calculation of non-invoiced completed jobs including applicable tax.

BIL2: Enables the user enter the payment mode (credit / debit card or check).

BIL3: Enables the user print or email invoices to customers.

BIL4: Provides for the calculation on interest and charges on overdue accounts or bounced checks.

BIL5: Enables the user confirm received payments.

### **Equipment Inventory**

EQU1: Provides access to equipment addition and deletion.

EQU2: Enables a user to change the working status of equipment.

EQU3: Enables a user locate equipment placement and period of use.



## 4.3 System Domain Models

### 4.3.1 Internal Domain Model

This is not a design document and so the model included here should not design all of the object interactions required, but provide an overview, that permits insights on

- (i) compatibility with object architectures and database architectures in other systems,
- (ii) data storage and data throughput requirements, and
- (iii) use-case completeness (making sure you have all the required use-cases).

Many experienced systems analysts produce a **data-flow diagram** supplemented with an ERD model in the next section, rather than a class-domain model, as this allows them to check for specification completeness in detail. Other analysts prefer to use a UML class-domain model. *I leave it to you which models you choose to include.* You need to consider issues (i) to (iii) above and explain the significance of these models to the specification.

### 4.3.2 Data Models

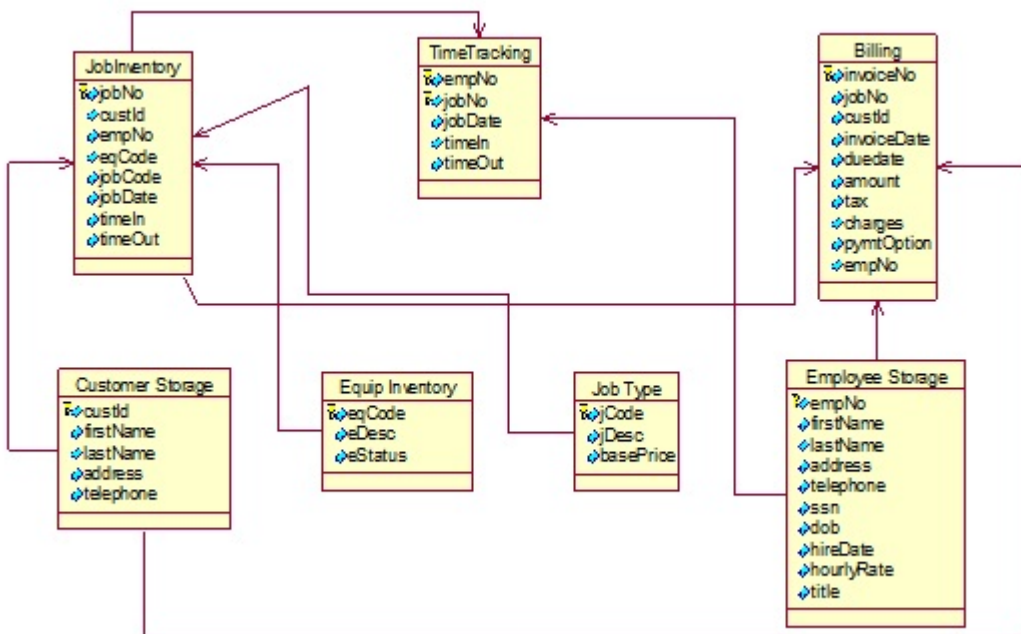


Figure 4.10 Data Model of PruneIt database

PruneIt's base (inputs) database will consist of 4 files. These will be customer storage that will contain all customer information, employee storage that will contain employee storage, equipment inventory that will contain equipment information, and job type that will contain information on job types and their base pricing. While most fields in these database files will require manual input, some data will be automatically generated and appropriate fields pre-filled. The envisioned database structures for each file are described below:

### Customer Storage

CustId: customer Id (auto generated) – char (4)  
firstName: Customer's first name – char (10)  
lastName: customer's last name – char (15)  
address: customer's home address – char (30)  
telephone: customer's telephone number – numerical (7)

### Employee Storage

EmpNo: employee number (auto generated) – char (6)  
firstName: employee's first name – char (10)  
lastName: employee's last name – char (15)  
address: employee's home address – char (30)  
telephone: employee's telephone number – numerical (7)  
ssn: employee's social security number – numerical (9)  
dob: employee's date of birth – date()  
hireDate: date when employee was hired -date()  
hourRate: employee's hourly pay rate – float (5.2)  
title: employee's title – char (10)

### Equipment Inventory

eqCode: equipment code (auto generated) – char (4)  
eDesc: equipment description – char (20)  
eStatus: status of equipment (hired / free / broken) – char (1)

### Job Type

jCode: job code (auto generated) – char (2)  
jDesc: job type description – char (20)  
basePrice: base pricing of job type – float (7.2)

On the basis of the base database files, 3 derived (outputs) databases files will be created that will enforce company business rules. These will include job inventory that will store all information jobs done or scheduled, time tracking that will store all information of landscaper hourly time inputs and billing that will contain all information customer invoicing. The derived database files will have some fields pre-filled, while others will have an auto-selection (combobox) feature to minimize input errors. Other fields will be manually input. The envisioned database structures for each file are described below:

### Job Inventory

jobNo: job number identification (auto generated) – numerical (6)  
custId: customer id related to job – char (4)  
empNo: employee number of assigned landscaper – linked list of char (6)  
eqCode: equipment code of assigned equipment – linked list of char (4)  
jobCode: job type to be undertaken – linked list of char (2)  
jobDate: date the job is to be undertaken – date()  
timeIn: time when job is started – time()  
timeout: time when job is completed – time()

### Time Tracking

empNo: employee number of landscaper – char (6)  
jobNo: job number identification – numerical (6)  
jobDate: date the job was undertaken – date()  
timeIn: time when job was started – time()  
timeout: time when job was completed – time()

### Billing

invoiceNo: invoice number (auto generated) – numerical (7)  
jobNo: job number identification – numerical (6)  
custId: customer id related to job – char (4)  
invoiceDate: date when invoice is generated – date()  
dueDate: date when invoice is supposed to be cleared (auto generated)  
date()  
amount: amount charged for the job done – float (9.2)  
tax: tax due on invoice amount – float (9.2)  
charges: extra charges due – float (9.2)  
pymtOption: payment option (credit card / debit card / check) – char (1)  
empNo: employee number of employee generating invoice – char (6)

## 4.4 Non-Functional Requirements

### 4.4.1 Usability

**NFR-1** PruneIt's interface is user-friendly and easy to get familiar with as it is based on the Microsoft User Interface guidelines..

**NFR-2** All users will be required to read the product's manual in order to understand all features.

**NFR-3** All user are required to be familiar with operating a PC in a windows environment. Users should be able to use PruneIt with maximum training of 2 hours. The expected education level of users should be of high school level.

**NFR-4** A context-sensitive help prompt should display when the user moves their mouse over any part of the screen. This prompt should appear within 500ms of the mouse movement, if the mouse is static for more than 300ms.

### 4.4.2 Reliability

This subsection specifies the following requirements associated with the reliability (e.g., mean time between failures, number of failures per unit time) of the system.

**NFR-5** Availability – The target goal of PruneIT is 100% available between the hours of 6 am- midnight, Monday to Saturday meeting the minimal needs of its end-users. Maintenance access should be between Midnight and 6 am Monday to Saturday and whole day Sunday. , degraded mode operations etc.

**NFR-6** Mean Time Between Failures (MTBF) – The mean time between failures shall exceed 6 months.

**NFR-7** Mean Time To Repair (MTTR) – for non-critical system failures is 30 minutes after contacting technical support within normal business hours (8am – 8pm). MTTR for critical system failures is 1 hour after contacting technical support during normal business hours.

**NFR-8** Accuracy – specify precision (resolution) and accuracy (by some known standard) that is required in the systems output.

**NFR-9** Maximum bugs or defect rate – Maximum Bugs have been established as .25 per 1000 lines of code providing for minimal coding error on delivered versions of PruneIt. This target can only been established after extensive testing.

**NFR-10** Bugs or defect rate – categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug (e.g., complete loss of data, complete inability to use certain parts of the functionality of the system).

### 4.4.3 Performance

This subsection specifies the following requirements associated with the speed with which PruneIt shall function.

#### 4.4.3.1 Capacity

This subsection specifies the following requirements concerning the minimum number of objects that PruneIt can support:

**NFR-11** The system shall support a minimum of 100 employees.

**NFR-12** The system shall support a minimum of 1,000 users.

**NFR-13** The system shall support a minimum of 10,000 simultaneous interactions.

#### 4.4.3.2 Latency

This subsection specifies the following requirements concerning the maximum time that is permitted for the system to execute specific tasks (i.e., system operations) or use case paths end to end:

**NFR-14** The typical search for any information should be within 5 seconds.

**NFR-15** The typical search for any information should be within 10 seconds for Mobile app.

#### 4.4.3.3 Response Time

This subsection specifies the following requirements concerning the maximum time that is permitted for PruneIt to respond to requests:

**NFR-16** All system responses shall occur within 30 seconds.

#### 4.4.3.4 Throughput

This subsection specifies the following requirements concerning how many executions of a given system operation or use case path must the system be able execute in a unit of time:

**NFR-17** Average of 10 transactions per second.

#### 4.4.3.5 Portability

This subsection specifies the following requirements associated with the ease with which PruneIt can be moved from one environment (e.g., hardware, operating system) to another.

PruneIt shall enable users to use the following environments (e.g., platform and operating system) to interact with:

User Personal Computer:

**NFR-18** PC with minimum of 2GB of RAM, 1.5 GHz

Operating Systems:

**NFR-19** Windows

**NFR-20** Macintosh

Mobile Device:

**NFR-21** Android

**NFR-22** Blackberry

**NFR-23** Windows Phone

**NFR-24** iPhone/iOS

#### **4.4.4 Supportability**

Software modifications should be envisioned in the first year of operation as the additional features are added to the system. All software updates will be subject to automated rigorous testing while version backups will be kept for a period of 12 months from the date when that version was installed.

## **5. SUPPLEMENTARY REQUIREMENTS**

### **5.1 Project management strategy**

The following table outlines the major features of PruneIt and assesses their priority, level of effort and risk.

<b>Feature</b>	<b>Priority</b>	<b>Effort</b>	<b>Risk</b>
Manage Customer Information	Critical	Medium	Medium
Manage Equipment and Jobs Data	Critical	Low	Medium
Generate Reports	Important	High	Medium
Cost/Profit Report	Important	Low	Low
Time Tracking Report	Important	Medium	Medium
Expense Report	Important	Medium	Medium

Inactive Customer Report	Useful	Low	Low
Job Report	Useful	Low	Low
Create Customer Invoices	Critical	High	High
Job Scheduling	Critical	High	High
Job Tracking	Critical	Medium	Medium
Time Tracking	Critical	High	Medium
Employee Scheduling	Important	Medium	Medium
View Company Schedule	Critical	Low	Low
Input Expenses	Critical	Medium	Medium

Project management will follow standard practices already established at the B.A.N.D. organization.

## 5.2 Systems Security and Audit

Systems security is a primal concern for B.A.N.D. Our clients are entrusting their confidential business and financial data to our software.

On the web tier, PruneIt will utilize Secure Sockets Layer (SSL) at all times. Access to the system will be granted only with a valid username and password combination. Users will be granted system roles to limit access to sensitive data within the application. All users will be granted roles by an administrator or managerial account.

The database will be housed internally and protected by a firewall. In addition, all data inputs will be thoroughly cleaned and validated before submitting to the data tier, so as to protect against SQL Injection attacks. Each customer's data will also be stored and protected by schema-level protection.

The database and web server will be subject to regular auditing. This auditing will consist of checking logs to identify "who" accessed "what" and "how" often. What actions were performed and what data was changed.

## 5.3 Assumptions and Dependencies

The initial user base of PruneIt is expected to be under 50 upon initial release. However, PruneIt is being developed as a generic application that can serve thousands of landscaping companies worldwide. Therefore the infrastructure must be able to scale to the appropriate level without losing quality.

As with any web application, a dependency is having a high speed Internet connection available. It is possible, however, that PruneIt may be unexpectedly disconnected from an

active connection at any time during it's use. Therefore, the application must always exit gracefully and any pending transactions should not be lost.

It is necessary to allow transactions initiated by a standard account to be overridden by an administrator account.

## 5.4 Requirements Traceability

In real life, this table would be very large – i.e. it would span several pages. For the purposes of this assignment, just take the first 3-4 features and map these to use-cases, which are then mapped to supplementary requirements. **Pay attention to this section, so that you understand how this is done – traceability is often critical to the success of requirements management, as the system design proceeds.**

Key: **F<sub>n</sub>** is High-Level System Feature **n** (from section 2.2.1).

**S<sub>n</sub>** is Non-Functional or Supplementary requirement **n** (from section 3.2).

### High-Level Features Mapped onto Use-Cases

Feature	UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8	UC 9	UC10
F1. Input and manage Customer information	X									
F2. Input and manage Employee information									X	
F3. Input and manage Equipment information										X
F4. Generate Reports			X		X					
F5. Create Customer Invoices				X						
F6. Input Job schedules						X			X	



F7. Input Job types									X	
F8. Track Landscaping Jobs		X	X		X		X	X		
F9. Input Daily Work on Jobs							X			
F10. View Employee Schedule					X					
F11. View Job information					X					
F12. View Company Schedule					X					
F13. Input Expenses			X							

**Use-Cases Mapped Onto Non-Functional and Supplementary Requirements**  
*(You may need to split this into one table for NF reqs. and another for supplementary reqs.).*

Use-Case	N FR -1	N FR -2	N FR -3	N FR -4	N FR -5	N FR -6	N FR -7	NF R- 8	NF R- 9	N FR - 10
UC1. Input and manage Customer information	X	X	X	X	X	X	X	X	X	X
UC2. Tracking Landscaping Jobs	X	X	X	X	X	X	X	X	X	X
UC3. Create Management Report	X	X	X	X	X	X	X	X	X	X
UC4. Create Customer invoice	X	X	X	X	X	X	X	X	X	X

UC5. Create Reports	X	X	X	X	X	X	X	X	X	X
UC6. Input Job Schedule	X	X	X	X	X	X	X	X	X	X
UC7. Input Daily Work on Jobs by Landscapers	X	X	X	X	X	X	X	X	X	X
UC8. Access PruneIt Information from Mobile Device	X	X			X	X	X	X	X	X
UC9. Input and manage Employee information	X	X	X	X	X	X	X	X	X	X
UC10. Input and manage Equipment information	X	X	X	X	X	X	X	X	X	X

## 6. Online User Documentation and Help System Requirements

### User Manual

B.A.N.D. will be creating a user manual to facilitate the basic operations of the system. The user manual will be available in pdf format for both online and offline access.

### Online Help

The user manual will be available online as well as a F.A.Q. page with most common questions/answers.

### Customer Support

B.A.N.D. will provide support to the customer during the initiation phase. This will include a demonstration and overview of the application, as well as help with loading existing customer, job, employee and equipment data.

## 7. Design Constraints

The system will run as a web application with access via an encrypted SSL login. PruneIt will run on all major web browsers - Internet Explorer, FireFox, Chrome, Safari and their mobile counterparts. We will customize key use cases to be available as mobile specific content; this could be challenging to make work on both tablets and phones. Additionally, each web browser adheres to web standards differently and PruneIt must overcome these differences in order to run seamlessly on any platform.

The system shall be developed using B.A.N.D.'s standard tool set for web applications - Java, Spring, MySQL, Apache and Windows.

## 8. Purchased Components

Software used to develop this product is either a) already purchased by B.A.N.D. or b) available as open source.

Hardware needed:

Dell PowerEdge T110 II (Web Server) ..... \$ 1300

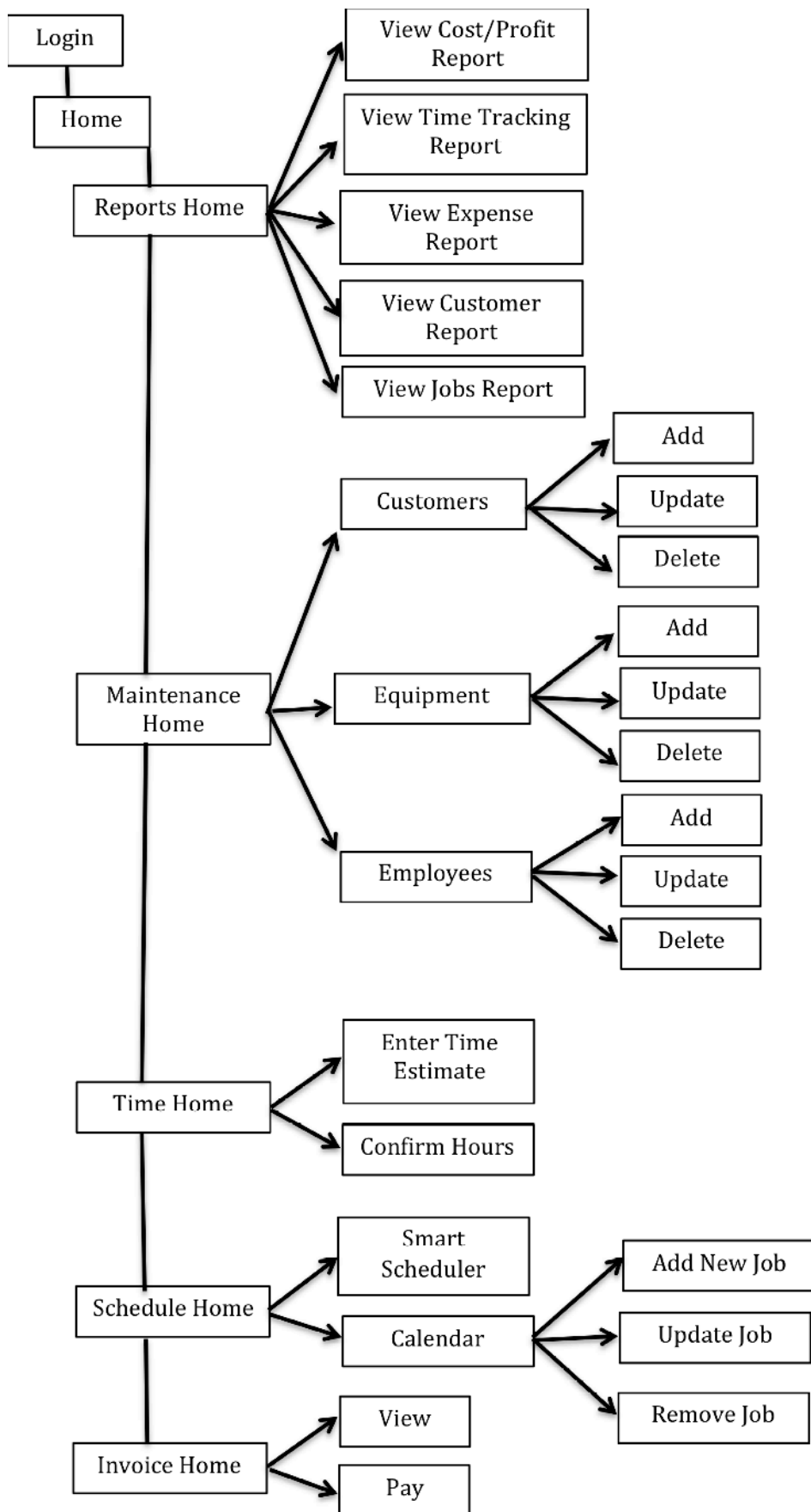
Dell PowerEdge T410 (Database Server) ..... \$ 1300

## 9. Interfaces

This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, etc, so that the software can be developed and verified against the interface requirements. *This section provides technical detail for the high-level view of these interfaces provided in section 2.*

### 9.1 User Interfaces

Screen Flow Diagram:



## **9.2 Hardware Interfaces**

The PruneIt web server will bind to the IP addresses of:

{Website TBD}

As more users/companies sign on to use PruneIt, more web servers may need to be added and load balanced.

TODO: add network diagram here?

## **9.3 Software Interfaces**

The web application server will communicate with the internal database server.

Subsequent version of PruneIt may support Calendar exports to iCal, Exchange and Google Calendar.

## **9.4 Communications Interfaces**

No communications interfaces are anticipated at this time.

## **10. Licensing Requirements**

The licensing for Microsoft Windows is included in the purchase of the hardware.

The licensing for the development software is distributed as follows:

Java is available under the GNU General Public License (GPL)

All Spring libraries are licensed under the terms of the Apache License, Version 2.0.

A MySQL commercial license is required.

Apache Tomcat is licensed under the terms of the Apache License, Version 2.0.

## **11. Legal, Copyright, and Other Notices**

PruneIt will be developed as proprietary software under the copyright notices of the B.A.N.D. corporation.

## **12. Applicable Standards**

B.A.N.D. follows industry standards for web development and user interface design. The PruneIt development team will follow in-house coding standards set by the B.A.N.D. corporation. All development standards are in-line with industry standards.

## ***Glossary***

PruneIt - The application under examination.

SDLC - Software Development Life Cycle

RUP - Rational Unified Process

SSL - Secure Sockets Layer

SQL - Structured Query Language