Group 15

Agilent Technologies

Customer Relations Management (CRM) System

Members:

Antoine Rocha
Ben Mitchell
Huanlei Wu
Haocheng Wang
John Gnanasekaran
Kyle Rovira
Kyung Yi
Pranav Raghavan

Table of Contents

Executive Summary

This documentation provides the planning, analysis, and evaluation of the supposed implementation of the Customer Relationship Management System. The Customer Relationship Management System connects the client to the company; whenever there is a service request from the client, the company attempts to find a fitting service engineer to accommodate the client's needs.

All parts of the system proposal can be found under the executive summary; calculations of profit can be found under the feasibility analysis in the economical feasibility section and calculations of the difficulty of the workforce and environment in the organizational feasibility section.

System Request

A document that describes the reasons for and the value added from building a new system
Contains 5 elements:

- **Project sponsor:** the primary point of contact for the project
    - Thomas Tong (thomas.tong@agilent.com) – Agilent Technologies
- **Business need:** the reason prompting the project
    - Clients are not able to report issues in the field well because there is no existing system to monitor field errors.
    - The organization needs a better way to organize everybody's time so that everyone can work on tasks that make the business most productive.
    - Some engineers are working more than others and it's hard for managers to track how hard they work.
- **Business requirements:** what the system will do
    - The system will manage the training, scheduling and maintenance of the service engineers who go out to client sites to install/repair/service equipment.
    - It will allow field managers to have oversight of service engineers in the workfield.
        - Each employee's record will be organized in the database.
        - Lets managers track company assets like equipment/products, courses offered.
        - Allows field managers to assign different client requests to the appropriate engineer.
    - Store data in rows about the various employees, equipment/products, courses and CRM.
    - Handles service orders and assigns proper service engineers to each order

    - Hierarchical control to allow managers to have power to reverse changes by service engineers on the databases.
    - Provide a notification when pertinent data changes
- **Business value:** How will the organization benefit from the project ?b
    - This system will allow the organization to allocate time and resources more efficiently.
    - Work distribution in the organization is more even
    - The system will allow for the balanced distribution of present and future work loads amongst qualified service engineers
- **Special issues:** Anything else that should be considered
    - CRM needs to make sure that multiple engineers do not receive the same service request
    - The system needs to make sure every service engineer gets an even amount of workload to avoid a skewed distribution of labor.

Work Breakdown Structure

I. System Modeling
   A. Inception
      1. Understand current business situation
         a) The client side issues
         b) Engineer side issues
         c) Manager side issues
   B. Elaboration
      1. Bring up the servers to host the data
      2. Write clients for
         a) Web-based application for Windows and Mac
         b) Android and iPhone apps for mobile access
   C. Construction
      1. Write the server side application to accept request from client, managers and engineers as well
      2. Client side application for employees as well as clients of the business.
   D. Transition
      1. Test the product and make sure that most of the obvious errors are removed before the product gets rolled out
   E. Production
      1. Install is very straightforward here so it would be the release date of the android and iOS stores
      2. Maintenance should take place like every product to fix bugs that come up after it has been rolled out to the users.

II. Requirements
    A. Inception
        1. Identify functional and nonfunctional requirements
        2. Analyze current system
        3. Create requirements definition
            a) Determine requirements to track
            b) Review requirements with sponsor
    B. Elaboration
    C. Construction
    D. Transition
    E. Production
III. Analysis
    A. Identify connections between Client's SO's and Service Engineer Deployments
    B. Identify required courses for qualifications of SE's
IV. Design
    A. Identify different classes
        1. Admin/Manager
        2. Employee
        3. Client
V. Implementation
    A. System 1: CRM
        1. Planning
        2. Analysis
        3. Design
        4. Implementation of the CRM system
    B. System 2: Employee records management
        1. Planning
        2. Analysis
        3. Design
        4. Implementation of the of the Employee records management system
    C. System 3: Employee skills management
        1. Planning
        2. Analysis
        3. Design
        4. Implementation of the Employee skills management system
    D. System 4: Employee learning system management
        1. Planning
        2. Analysis
        3. Design

4. Implementation of the Employee learning system management system

VI. Test

A. Testing of the entire system, as a whole.

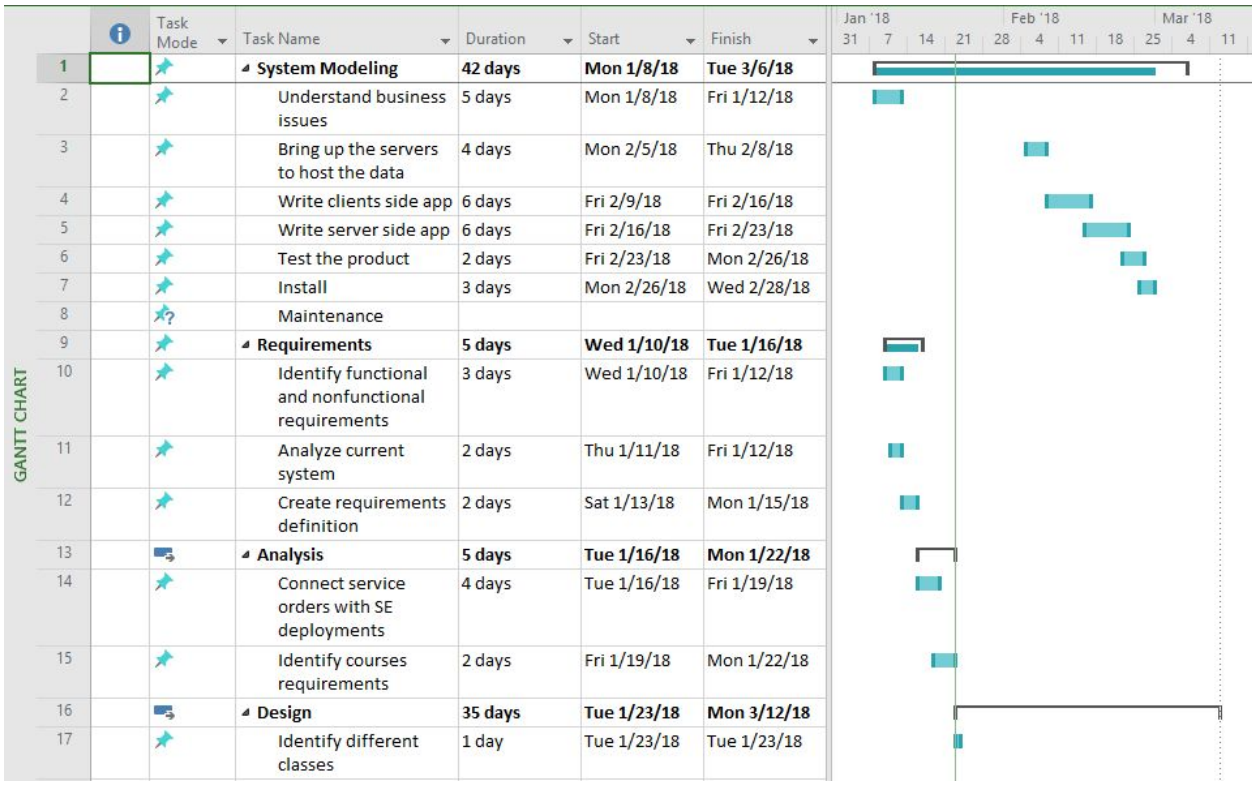
VII. Deployment

A. Install the system on the different Vans

B. Install the system on the manager's computers

C. Analyse the errors that come up from being deployed in the field

Gantt Chart

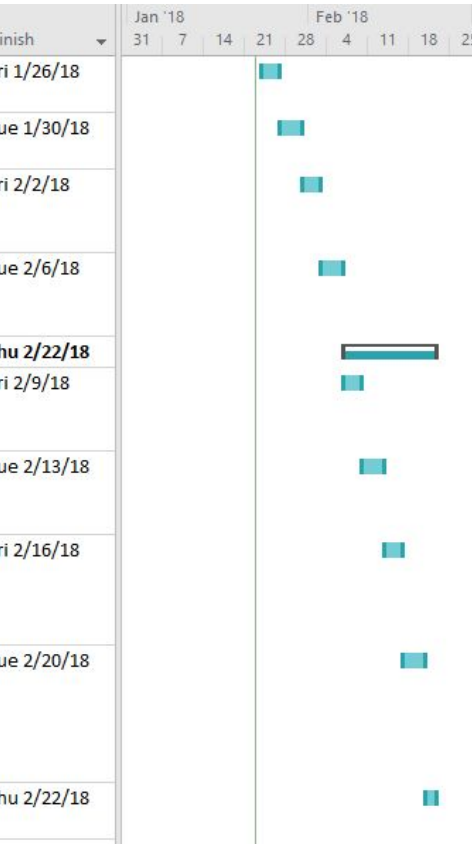


GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish
1		**System Modeling**	**42 days**	**Mon 1/8/18**	**Tue 3/6/18**
2		Understand business issues	5 days	Mon 1/8/18	Fri 1/12/18
3		Bring up the servers to host the data	4 days	Mon 2/5/18	Thu 2/8/18
4		Write clients side app	6 days	Fri 2/9/18	Fri 2/16/18
5		Write server side app	6 days	Fri 2/16/18	Fri 2/23/18
6		Test the product	2 days	Fri 2/23/18	Mon 2/26/18
7		Install	3 days	Mon 2/26/18	Wed 2/28/18
8		Maintenance			
9		**Requirements**	**5 days**	**Wed 1/10/18**	**Tue 1/16/18**
10		Identify functional and nonfunctional requirements	3 days	Wed 1/10/18	Fri 1/12/18
11		Analyze current system	2 days	Thu 1/11/18	Fri 1/12/18
12		Create requirements definition	2 days	Sat 1/13/18	Mon 1/15/18
13		**Analysis**	**5 days**	**Tue 1/16/18**	**Mon 1/22/18**
14		Connect service orders with SE deployments	4 days	Tue 1/16/18	Fri 1/19/18
15		Identify courses requirements	2 days	Fri 1/19/18	Mon 1/22/18
16		**Design**	**35 days**	**Tue 1/23/18**	**Mon 3/12/18**
17		Identify different classes	1 day	Tue 1/23/18	Tue 1/23/18

GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish
18		Design learning system management	3 days	Wed 1/24/18	Fri 1/26/18
19		Design employee skills managements	3 days	Sat 1/27/18	Tue 1/30/18
20		Design Employee records management schema	3 days	Wed 1/31/18	Fri 2/2/18
21		Design customer relationship management system	3 days	Sat 2/3/18	Tue 2/6/18
22		**Implementation**	**12 days**	**Wed 2/7/18**	**Thu 2/22/18**
23		Implement learning system management	3 days	Wed 2/7/18	Fri 2/9/18
24		Implement employee skills managements	3 days	Sat 2/10/18	Tue 2/13/18
25		Implement Employee records management schema	3 days	Wed 2/14/18	Fri 2/16/18
26		Implement customer relationship management system	3 days	Sat 2/17/18	Tue 2/20/18
27		Training Employees	2 days	Wed 2/21/18	Thu 2/22/18
28		**Testing**	**7 days**	**Fri 2/23/18**	**Mon 3/5/18**
29		Test all the systems	7 days	Fri 2/23/18	Mon 3/5/18
30		**Deployment**	**5 days**	**Tue 3/6/18**	**Mon 3/12/18**
31		Deploy all the systems	5 days	Tue 3/6/18	Mon 3/12/18



Feasibility Analysis

Technical Feasibility

This project carries a high level risk due to the lack of expertise and experience from working on a similar project. But the system itself will be started from scratch, as there is no pre-existing system. This means the new system will be easier to integrate since there exists no prior system to have to integrate with. User-wise compatibility is uncertain although very flexible considering the web based servers than can run on any kind of operating system. Further involvement will be necessary for concept, testing, and further requirements determination In terms of project size, this carried medium risk since there are fewer than ten people.

Economical Feasibility

Based on the cost benefit analysis shown below, this project could lead to greater net business profit through increased sales and better customer service. The development of the system itself would be a one time expenditure with follow up maintenance. Operation costs would be minimal, because of the Web-Based Server for the system. The Web-Based Server allows there to be no additional equipment required. Thus if the project leads to a modest increase in the number of sales, then the business will see a substantial increase in profit after the first year. Intangible benefits include improved customer service due to the system's ease of use and higher employee satisfaction due to balanced and efficient workloads. A proper CRM system can be expected to raise sales by 30%, even with a conservative estimate of a 15% increase for a first time implementation, which is a substantial payoff.

	2018	2019	2020	Total
Benefit				
Increased Sales	$4,000,000	$4,000,000	$4,000,000	$12,000,000
Customer Service	$500,000	$450,000	$400,000	$1,350,000
TOTAL BENEFIT:	$4,500,000	$4,450,000	$4,400,000	$13,350,000
PV BENEFIT:	$4,500,000	$4,278,846	$4,068,047	$12,846,893
PV OF ALL BENEFITS	$4,500,000	$8,778,846	$12,846,893	$26,125,740
Cost				
Labor: Analysis and design	$200,000	$5,000	$3,000	$208,000
Labor: Implementation	$330,750	$20,000	$20,000	$370,750
Development training	$330,750	$33,000	$33,000	$396,750
Hardware	$80,000	$50,000	$50,000	$180,000

Software	$15,000	$543,000	$543,000	$1,101,000
TOTAL DEVELOPMENT COST:	$956,500	$651,000	$649,000	$2,256,500
Labor: Management oversight	$350,000	$350,000	$350,000	$1,050,000
Software Upgrades	$200,000	$200,000	$200,000	$600,000
Operational Team Salaries	$375,000	$375,000	$375,000	$1,125,000
User training	$202,500	$100,000	$100,000	$402,500
TOTAL OPERATIONAL COST:	$1,127,500	$1,025,000	$1,025,000	$3,177,500
TOTAL COSTS	$2,084,000	$1,676,000	$1,674,000	$5,434,000
PV COSTS	$2,084,000	$1,676,000	$1,674,000	$5,434,000
PV OF ALL COSTS	$2,084,000	$3,760,000	$5,434,000	
TOTAL PROJECT BENEFITS/COST:	$2,416,000	$2,774,000	$2,726,000	$7,916,000
Yearly NPV:	$2,416,000	$2,602,846	$4,068,047	$9,086,893
Cumulative NPV:	$2,416,000	$5,018,846	$9,086,893	
ROI				167.22%
Break Even Point.				1 year(s)

Organizational Feasibility

From an organizational perspective, this project has low risk. The goals of the system is to improve efficiency of communication both for clients to report field errors easier and also within the company to better organize time and personnel. Since this is the first system of its kind to be implemented, user acceptance is projected to be high. Furthermore, because the organization is already divided into different positions, these positions can easily be integrated as levels of user access and interaction. The system aligns with business goals well to increase productivity as better management of operations and personnel will lead to increase in productivity in clients, managers, and engineers at all levels.

Unadjusted Actor Weighting Table:				
Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with Well Defined API	1		
Average	External System with	2		

	Protocol Interface			
Complex	Human	3		
				Unadjusted Actor Weight Total (UAW)

Unadjusted Use Case Weighting Table:

Use Case Type	Description	Weighting Factor	Number	Result
Simple	1-4 orders	5		
Average	5-9 orders	10		
Complex	>9 orders	15		
				Unadjusted Use Case Weight Total (UUCW)

Unadjusted use case points (UUCP) = UAW + UUCW

Technical Complexity Factors:

Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
T1	Distributed System	2	0	0
T2	Response time or throughput performance objectives	1	0	0
T3	End-user online efficiency	1	3	3
T4	Complex Internal Processing	1	0	0
T5	Code Reusability	1	1	1
T6	Ease of Installation	0.5	1	0.5
T7	Ease of Use	0.5	2	1
T8	Portability	2.0	4	8
T9	Ease of Change	1.0	2	2
T10	Concurrency	1.0	0	0
T11	Special security objectives included	1.0	0	0
T12	Direct access for third parties	1.0	1	1

T13	Special user training required	1.0	1	1
				Technical Factor Value (TFactor): 17.5
Technical complexity factor (TCF) = 0.6 + (0.01 * TFactor) **0.6+(.175) = .775**				
Environmental Factors:				
Factor Number	Description	Weight	Assigned Value (0-5)	Weighted Value
E1	Familiarity with system development process being used	1.5	0	0
E2	Application experience	0.5	0	0
E3	Object-oriented experience	1.0	2	2
E4	Lead analyst capability	0.5	0	0
E5	Motivation	1.0	4	4
E6	Requirements stability	2.0	2	4
E7	Part-time staff	-1.0	0	0
E8	Difficulty of programming language	-1.0	4	-4.0
				Environmental Factor Value (EFactor): 6
Environmental factor (EF) = 1.4 + (-0.03 * EFactor) = 1.22 Adjusted use case points (UCP) Person hours multiplier (PHM) Person hours				

Requirements Definition

Customer Relations Management Subsystem

The application is the main platform that the clients will interact with the business. The client is able to create a service order. Then the system will search through the Employee Skills Managements scheme to find an engineer with the necessary qualifications and skill set. If the engineer is found, the system will schedule a service engineer (SE) for the service order (SO). The system will then alert the client the service is underway. Else, if no engineer is found, the system will send an alert to a manager if no service engineer is found. Once the service order has been completed, the system will mark as complete and archive record.

Functional Requirements

Manage Service Orders:

1. Client creates service order.
2. Client changes service order request.
3. Client cancels service order.

Manage Service Engineers and SE schedules:

1. Update qualifications of employed SEs
2. Assign qualified SEs to specific SOs
3. If an SE becomes unavailable, replace with another qualified SE
4. Keep track of SE schedules/appointments

Record history on employees, clients, and SO's:

1. Manage course credits of employed Service Engineers
2. Manage personal information of past and present clients
3. Keep track of completed service orders

Alert Clients about SO

1. Alert client if SE found
2. Alert Client if SE not found and explain why

Nonfunctional Requirements

Operational:

1. The system will have to operate online
2. Website has to be secure and live
3. Clients have to be able to fill out and create new service orders

Performance:

1. Company sends email when SO is received
2. Company must not take > 3-5 business days to respond to client about receiving SO
3. Time taken to find SE must not take more than 5 days
    a. if more than 5 days, alert client

Security:

1. Vans will have real-time GPS signal to indicate where vehicle is at
2. Only SE and higher-ups will have access to client SO info

Reliability:

(Failures include: website crashing, incorrect links, clients unable to submit form)

1. Website maintenance every month for 12 hours to avoid bugs
2. Page dedicated for user complaint about website failures
3. Special email inbox for complaint of unreceived packages

Cultural and Political Influences:

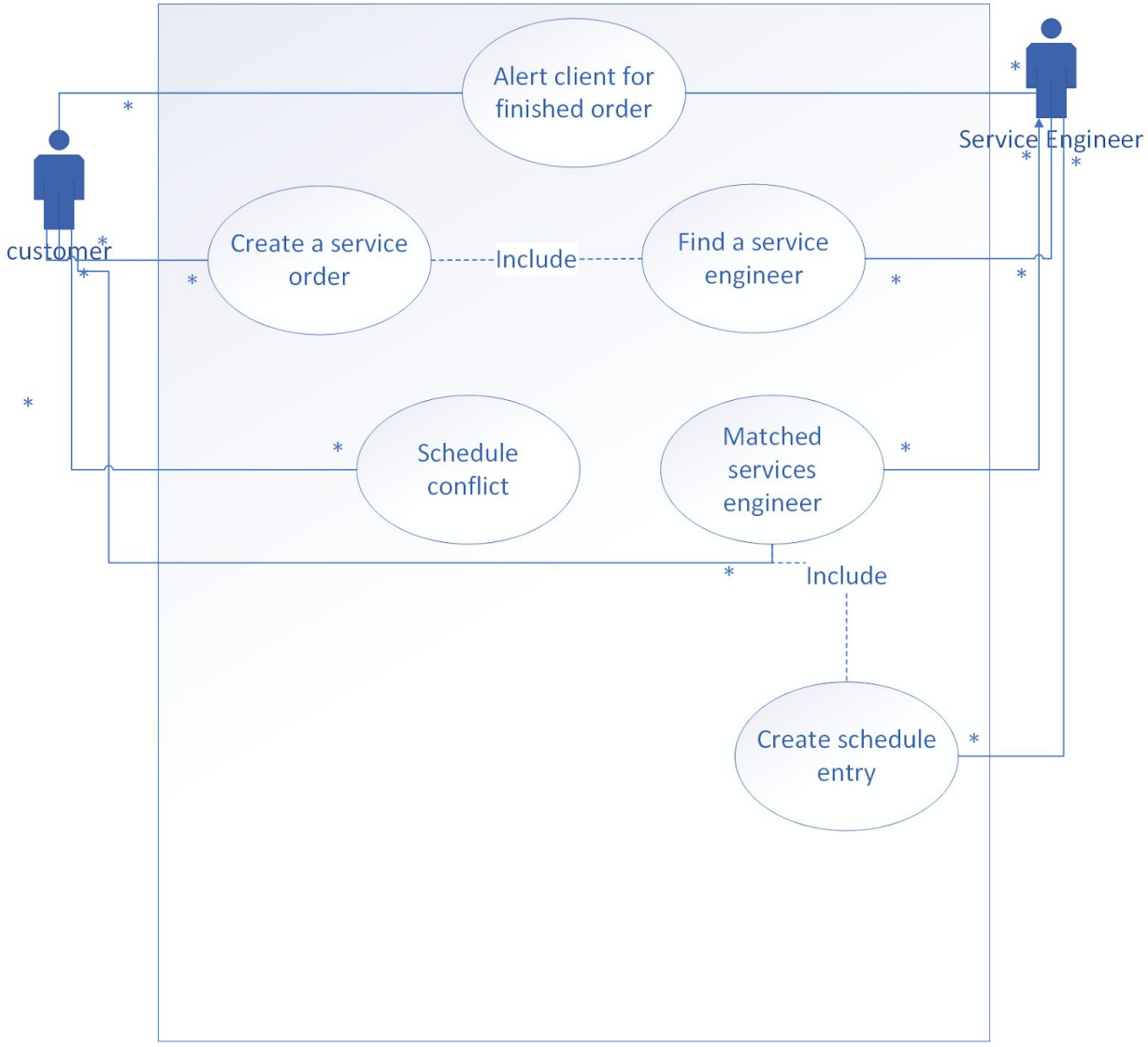
1. Country managers will be able to define new fields in product database to capture country-specific information
2. Country managers will be able to change the format of the telephone number field in the customer database
3. All platforms will contain all special characters of all languages (unicode)

   Legal Implications:

   1. Client confidentiality is protected by US Law
   2. Located in California: both parties must know that they are going to be recorded in order for recording to proceed

Functional Model

Use-Case Diagram



Use Case Descriptions

Use Case Name: Request a Service	**ID** – #00001	**Importance Level**: High
Primary Actor: Client		**Use Case Type**: Detailed, Essential
Stakeholders and Interests: Client - wants to request a service		
Brief Description: Client wants to request a service, and system creates a Service Order		
Trigger: When the client first requests an order **Type**: External		
Relationships: Association: Client Include: Extend: Generalization:		
Normal Flow of Events 1. Client logs into account 2. Client fills out claim a. Client provides information with equipment, and nature of problem 3. Client confirms address of service order 4. If Client information has changed a. Execute the Update Client Account use case 5. Client submits service request to the system 6. System creates a Service Order 7. Client receives order number		
SubFlows: S-1: New Service Order 1. System asks Client for nature of problem and equipment. 2. System asks Client for possible work times. S-2: Cancel Service Order 1. System asks Client for order number. 2. System finds service order and cancels it. S-3: Change Service Order 1. System performs S-2: cancel service order. 2. System performs S-1: new service order.		
Alternate/Exceptional Flows:		

Use Case Name: Search for a suitable Service Engineer for new Service Order	**ID** – #00002	**Importance Level**: High
Primary Actor: Service Order		**Use Case Type**: Detailed, Essential
Stakeholders and Interests: Client - Needs a service engineer to fulfill their request Service Order - Searches for Service Engineer by matching with Employee Skills Information		
Brief Description: This use case describes the Service Order for a new suitable engineer.		
Trigger: Request file for new service engineer **Type**: Internal		
Relationships: Association: Client, Service Engineer, Employee Skills Info Include: Extend: Generalization:		
Normal Flow of Events 1. Identify the type of problem in the system to identify the type of engineer required 2. Go through the list of available engineers when the client wants to receive our staff and pick 2 engineers for the job. One engineer is on standby in case the first engineer gets caught up with one of the other service tasks. 3. If no engineer can be found a. Execute the Alert when no SE is found use-case 4. If a suitable engineer is found a. Execute the Schedule SE use-case		
SubFlows:		
Alternate/Exceptional Flows:		

Use Case Name: Alert when no SE is found for the SO	**ID** – #00003	**Importance Level**: High
Primary Actor: Client		**Use Case Type**: Detailed, Essential
Stakeholders and Interests: Clients - Needs to get notified when an Engineer is not found for his/her service order		

Brief Description: This use case describes what happens when a SE is not found
Trigger: No Service engineer is available for a specific service order **Type**: External
Relationships: Association: Client, engineer Include: Extend: Generalization:
Normal Flow of Events 1. System receives a service order from client 2. Use the information provided, the system will try to search for a suitable service engineer to fulfill this order. 3. If a service engineer cannot be found, notify the client
SubFlows:
Alternate/Exceptional Flows:

Use Case Name: Alert client when SE is assigned to SO	**ID** – #00004	**Importance Level**: High
Primary Actor: Administrative Staff		**Use Case Type**: Detailed, Essential
Stakeholders and Interests: Administrative Staff - wants to notify client about a SE being assigned to their respective SO Clients - waiting for notification of a Service Engineer being assigned		
Brief Description: This use-case describes what happens once a service engineer is assigned to a service order		
Trigger: An engineer is being assigned **Type**: External		
Relationships: Association: Client, Service engineer Include: Extend: Generalization:		

Normal Flow of Events:
1. A suitable service engineer has been found for the SO
2. The service engineer is assigned to the service order in question
3. Client is contacted and notified about the appointment

SubFlows:

Alternate/Exceptional Flows:

Use Case Name: Schedule SE	**ID** – #00005	**Importance Level**: High
Primary Actor: Administrative Staff		**Use Case Type**: Detailed, Essential

Stakeholders and Interests:
Administrative Staff - contacts engineer to schedule them for SO
Service Engineer - waiting to be assigned for a service order

Brief Description:
This use-case describes what happens

Trigger: An engineer needs to be assigned to a order

Type: External

Relationships:
Association: Service engineer
Include:
Extend:
Generalization:

Normal Flow of Events:
1. System found an available service engineer for a service order
2. The service engineer is notified and offered the job
    a. If the engineer is interested and wants the job
        i. The S-1: Assign engineer subflow is performed
    b. If the engineer refuses the job
        i. The S-2: find alternate engineer subflow is performed

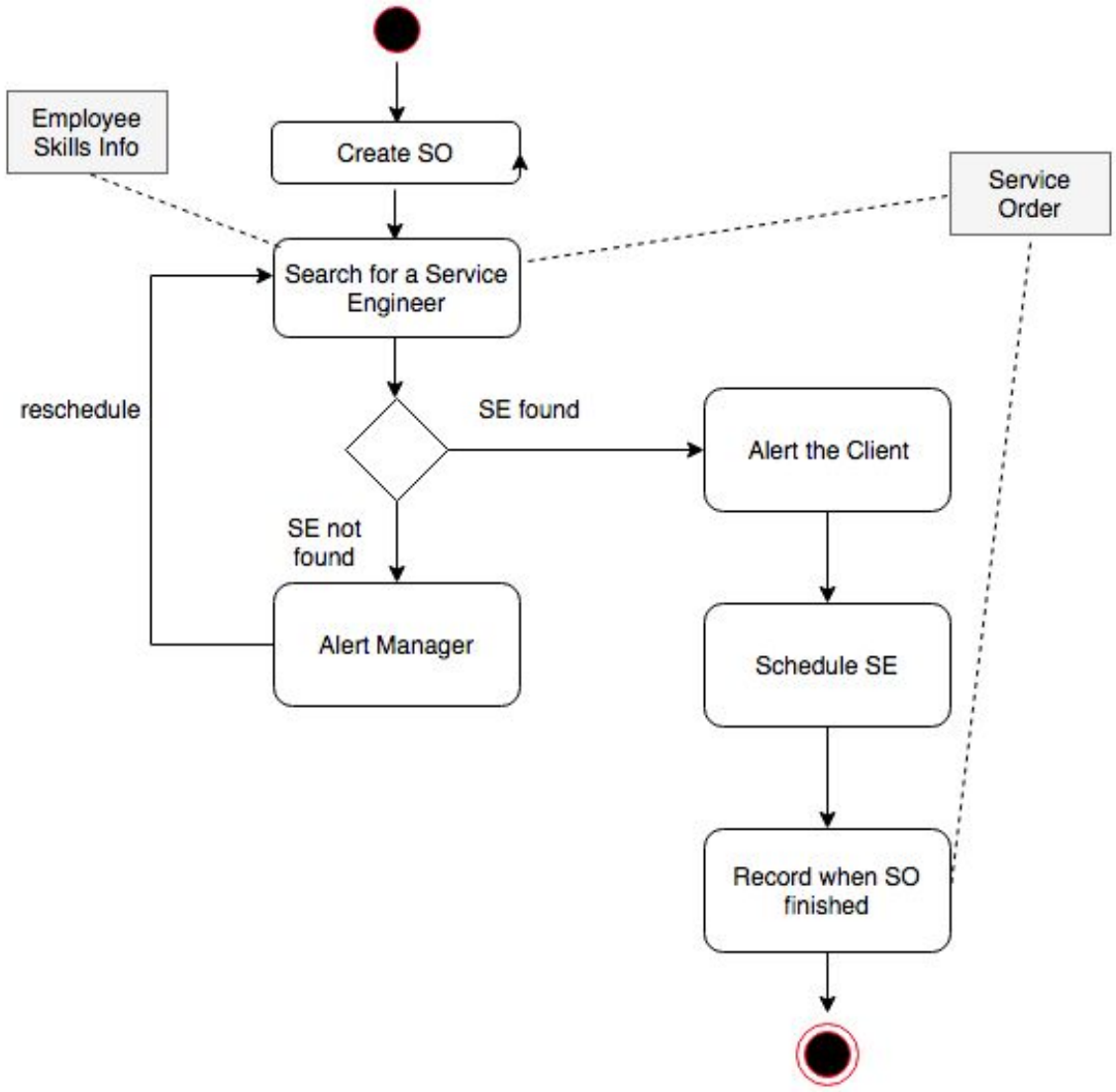
SubFlows:
S-1: Assign Engineer
1. Engineer agrees to a time and place for the SO
2. Execute Alert client when SE is assigned to SO use-case
3. Engineer conducts SO

S-2: Find alternate engineer
1. Contact reserve engineer for assignment to SO

2. If reserve engineer accepts, perform S-1 subflow 3. If reserve engineer refuses, execute the Search for a suitable Engineer for new Service Order use-case
Alternate/Exceptional Flows:

Use Case Name: Record when the SO has been serviced	**ID** – #00006	**Importance Level** – High
Primary actor: Client		**Use case type**: Detail, Essential
Trigger: When the client has been serviced the client rep will notify the system and we will begin processing the receipt that is emailed to them.		
Type: External		
Relationships Association: Client, Service engineer Include Extend Generalization		
Normal flow of events: 1. Client's issue has been handled by the service engineer so the order can be marked as completed 2. The accounts team processes the expenditures that he may be liable for 3. The information provided from the accounts team can be processed and delivered to the client by email.		
Subflow: S-1: 1. The search for the service engineer was not found so the client is notified that he might have to chose different times to schedule the visit from the engineers. S-2: 1. If the engineer has visited the client and is still not able to fix the issue that was initially brought up, a manager is notified. 2. Another order needs to be created, but a description of what happened needs to be processed for the client.		
Alternate/Exceptional Flow:		

Activity Diagram



Structural Model

CRC Cards

Class Name: Employee Skills Info	**ID:** 0001	**Type:** Concrete
Description: When the service order comes in, the employee skills management system needs to go through it and come up with skills that a potential engineer that services the product might need.		**Associated Use-Cases:** 2
Responsibilities ● Include skills that are within the reach of employee ● Take input from the employee skills system	**Collaborators** Service Order Employee Skills Management System	
Attributes ● Skills array ● Service note string ● Service order number ● Times available		
Relationships Generalization: Thing Aggregation: Other Associations: Service Order		

Class Name: Service Order	**ID:** 0002	**Type:** Concrete, Domain
Description: The service request that is submitted by the client online.		**Associated Use-Cases:** 1, 3, 5
Responsibilities ● Contains information about the service order ● Allows the employee system to find out the skills required for the service	**Collaborators** Service Order Employee Skills Info	
Attributes ● service order number ● employee badge numbers ● skills required ● skills found		

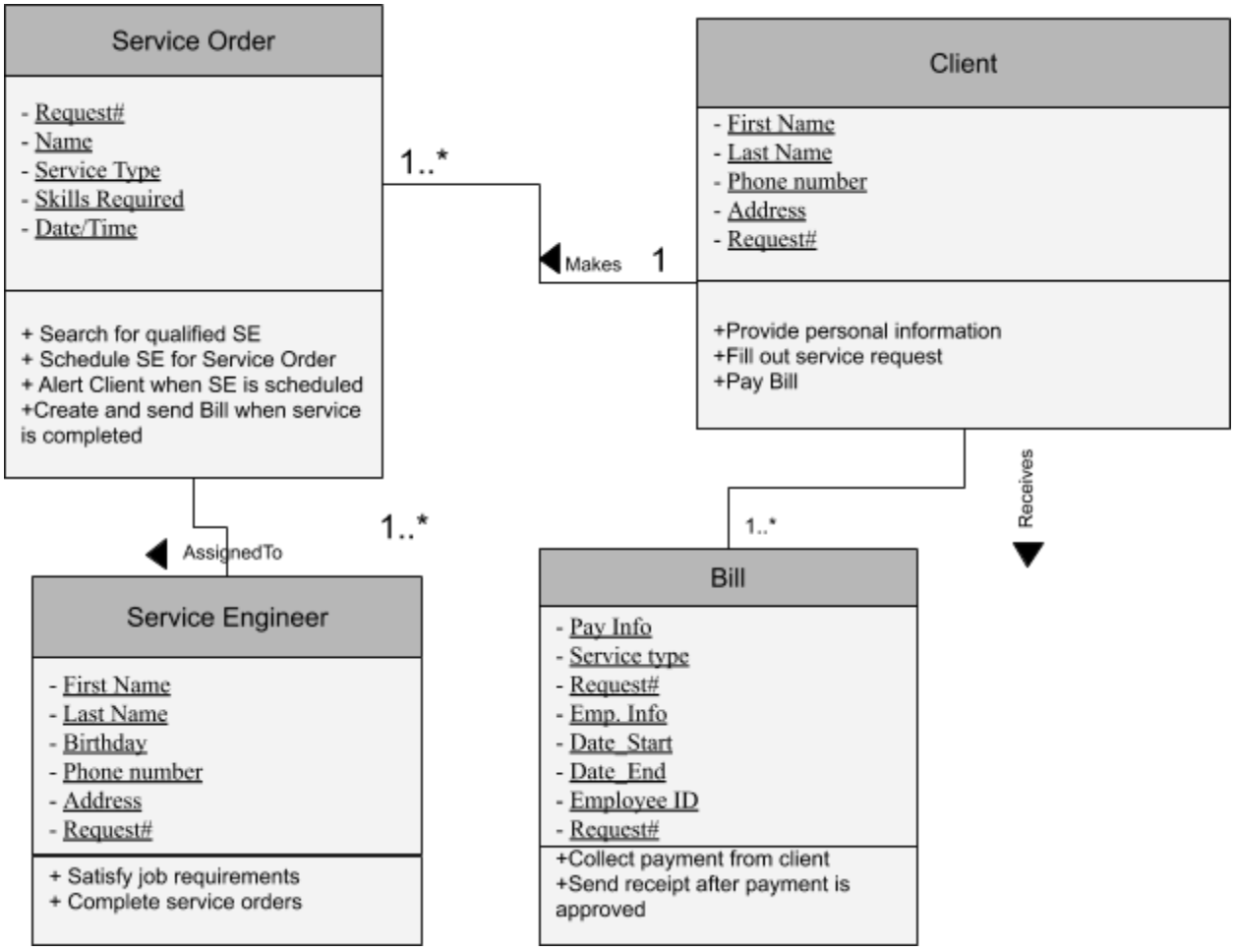
• completed (0 or 1) • detailed description of issue • times available
Relationships Generalization: Thing Aggregation: None Other Associations: Client, Service Engineer, Employee Skills Info

Class Name: Client	**ID:** 0003	**Type:** Concrete, Domain
Description: An individual who requires assistance with troubleshooting and fixing their IT systems		**Associated Use-Cases:** 2, 4
Responsibilities • Make Service Requests • Make Payments • Get Receipts	**Collaborators** Service Order Payments Receipts	
Attributes • Name • Phone Number • Address • SO#		
Relationships Generalization: Person Aggregation: None Other Associations: Service Order, Bill		

Class Name: Bill	**ID:** 0004	**Type:** Concrete, Domain
Description: Informs client about service request amount and records client's payment info		**Associated Use-Cases:** 3
Responsibilities n/a	**Collaborators** n/a	
Attributes ● Total amount ● Payment type ● Service type ● Employee name ● Start date ● End date ● Request number		
Relationships Generalization: Thing Aggregations: None Other Associations: Client		

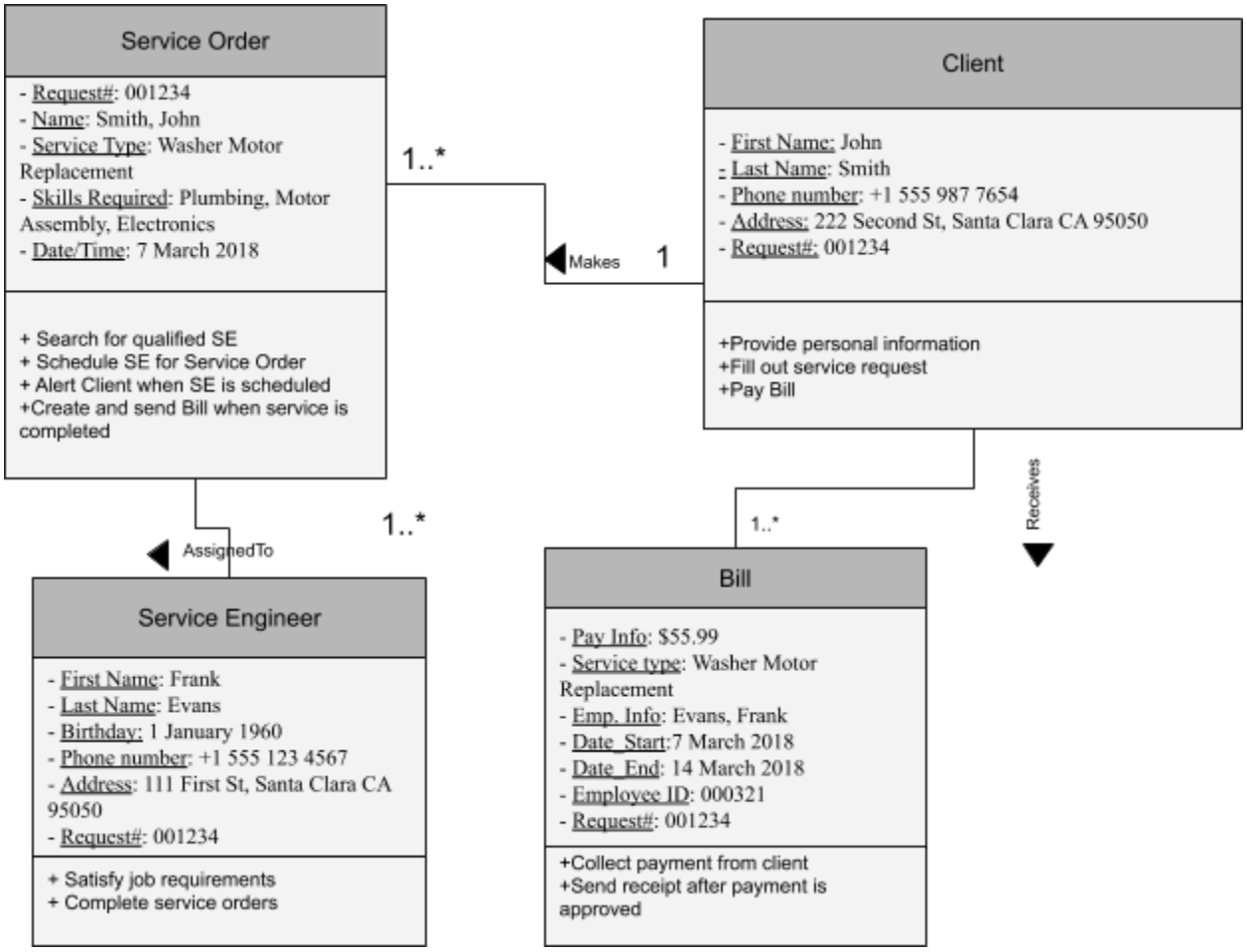
Class Name: Service Engineer	**ID:** 0005	**Type:** Concrete, Domain
Description: An individual who receives a service order and ensures its completion		**Associated Use-Cases:** 2
Responsibilities ● Complete Service Orders	**Collaborators** Service Order	
Attributes ● Name ● Birthday ● Phone Number ● Address ● SO#		
Relationships Generalization: Person Aggregation: Other Associations: Service Order, Reminders		

Class Diagram



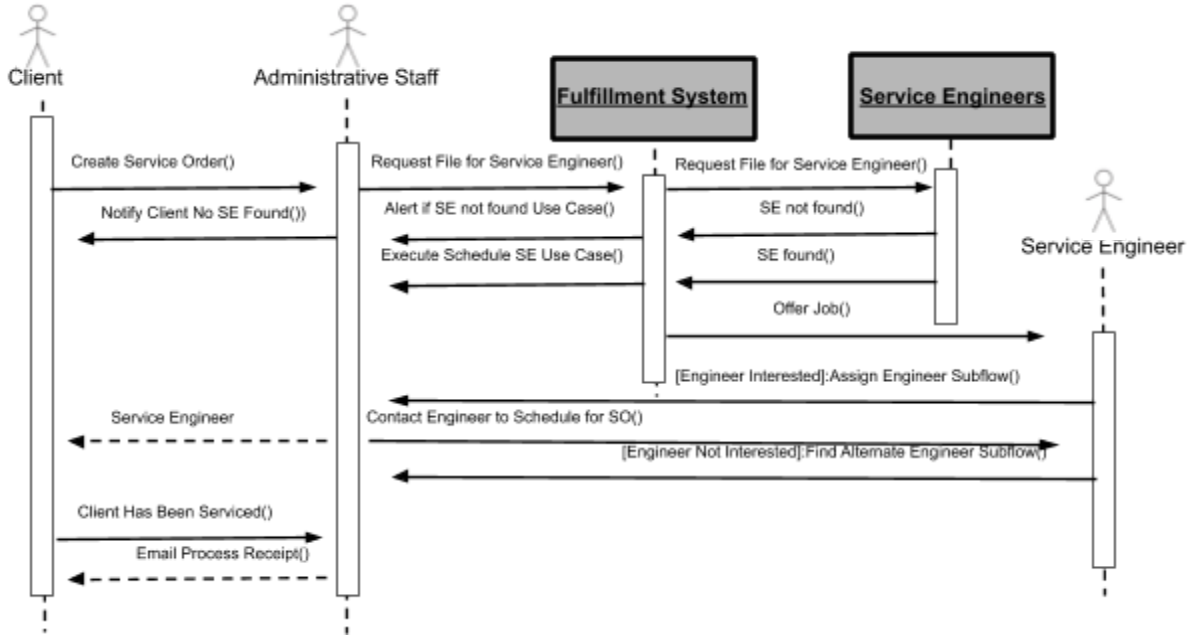
Object Diagram

Service Order

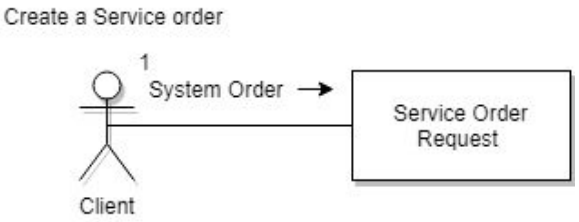


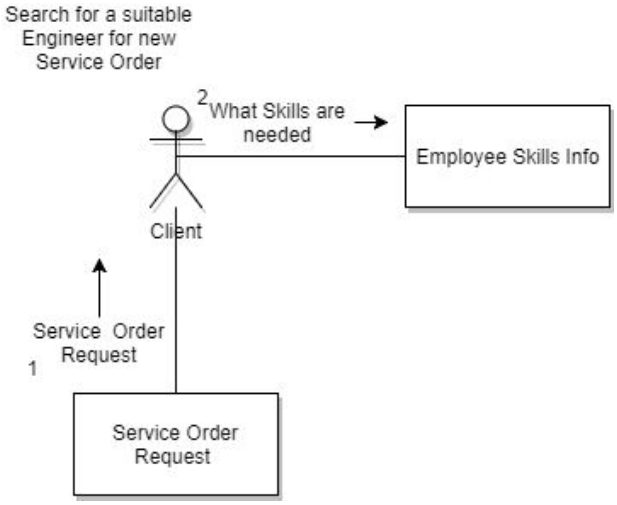
Behavioral Model

Sequence Diagram

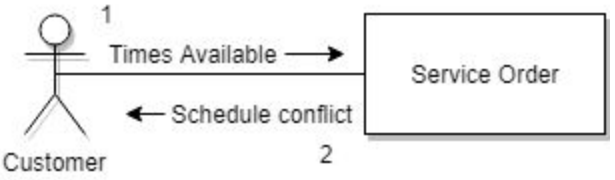


Communication Diagram

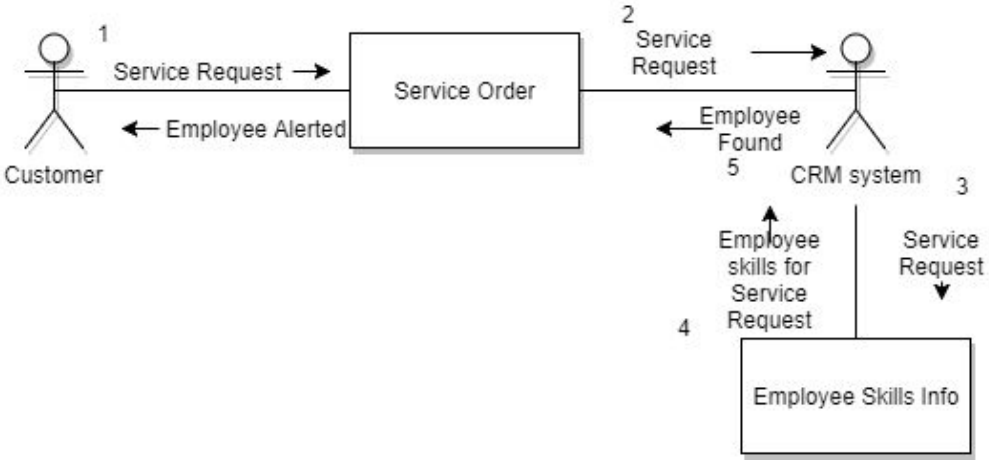




Alert when no S.E is found for the S.O

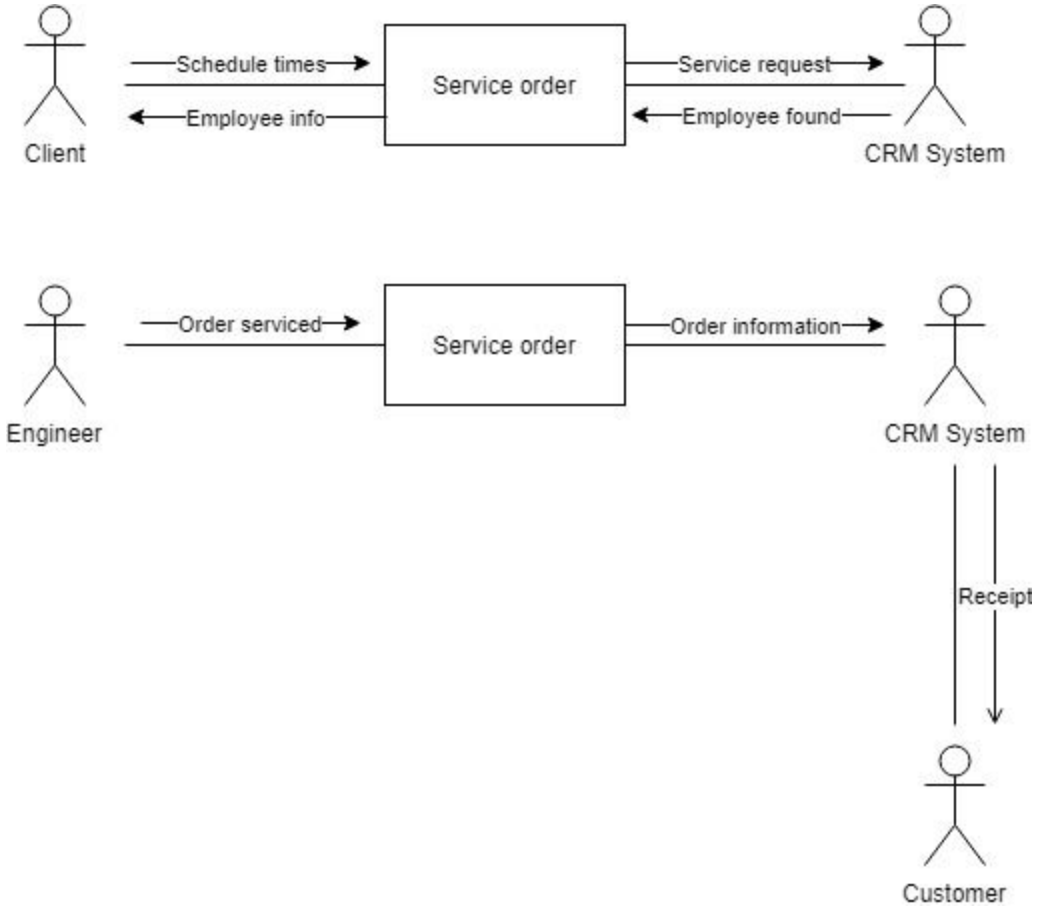


Alert client when S.E is assigned to S.O

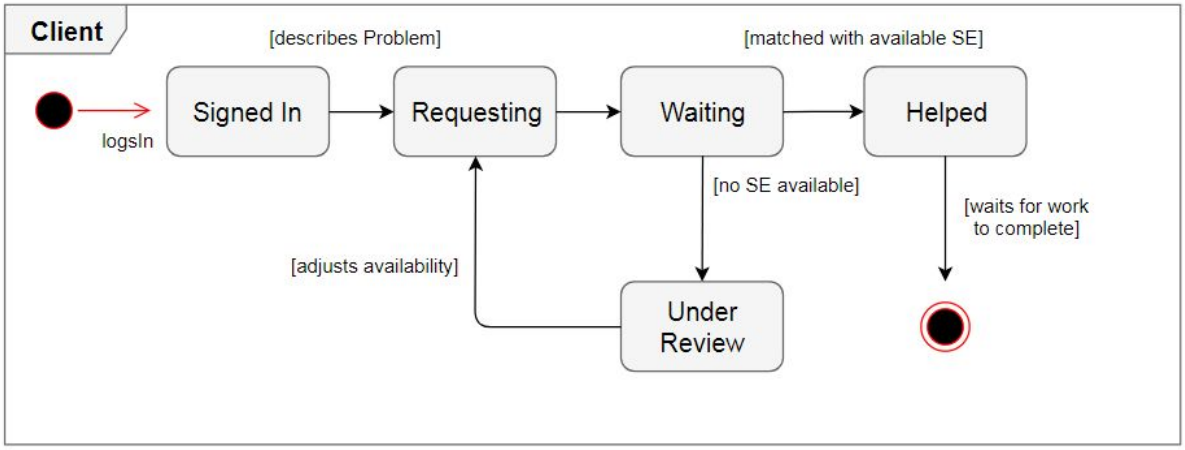


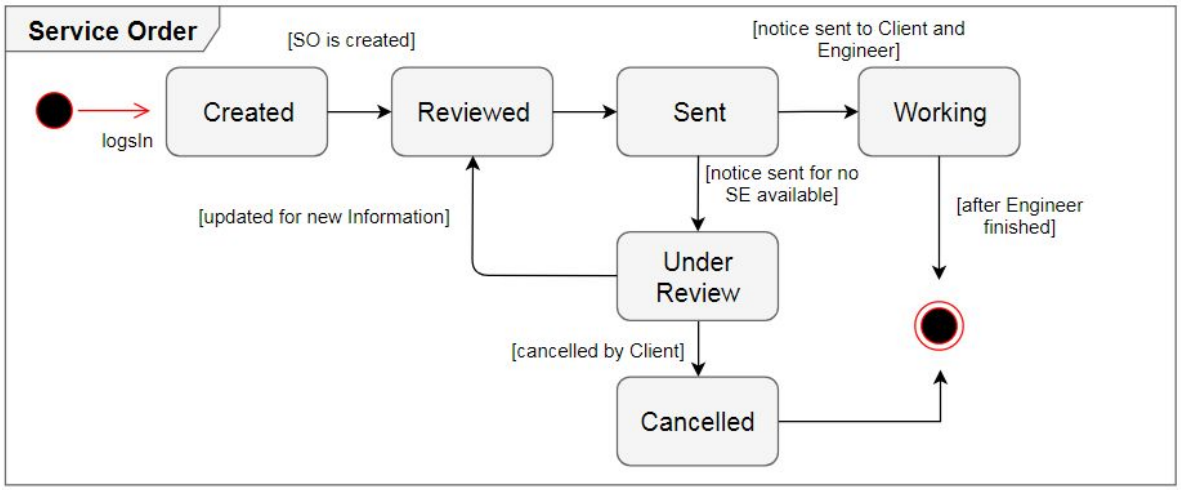
Alert Client when engineer is not found

Record when service order has been serviced.



Behavioral State Machines



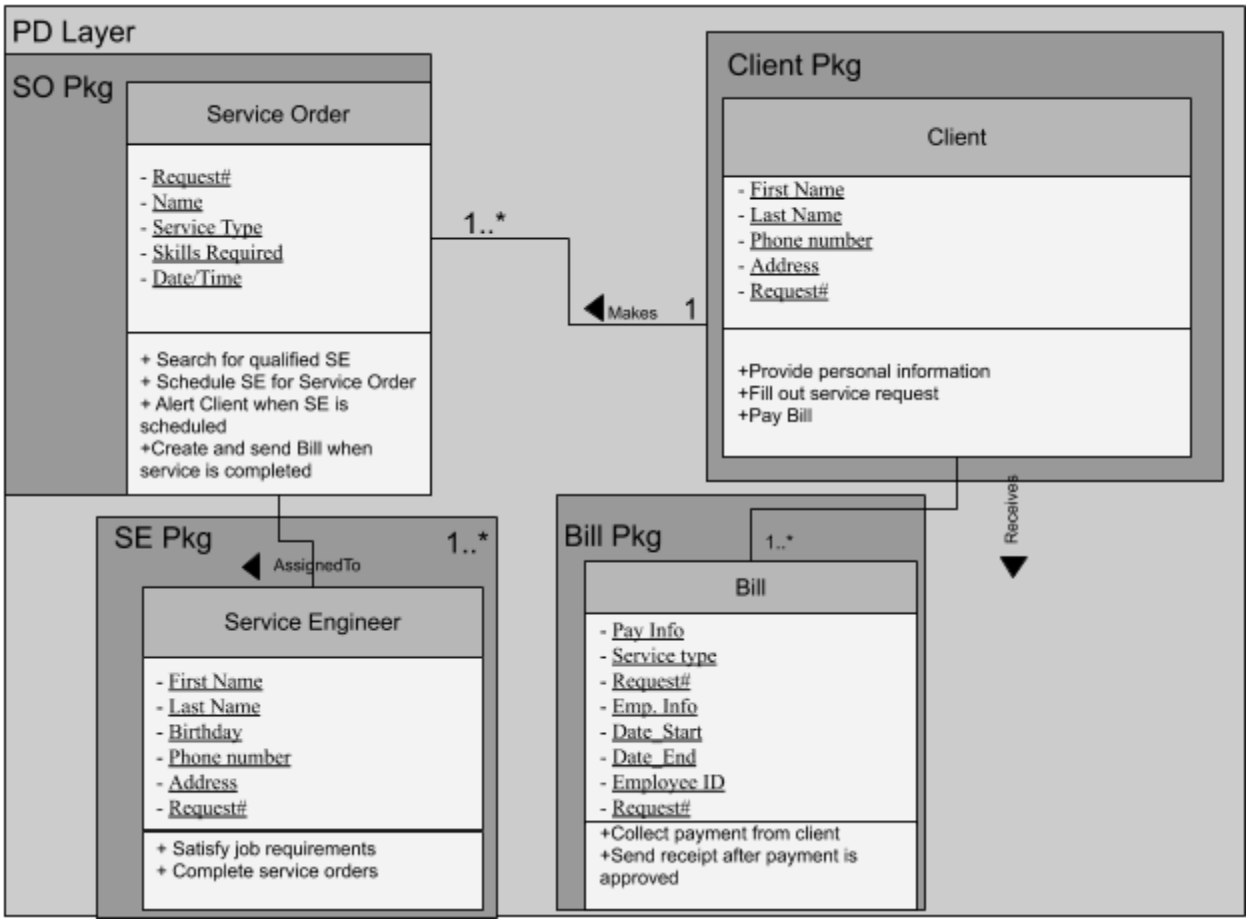


CRUDE Matrix (Create, read, update, destroy, execute)

	Client	Service Engineer	Service Order	Bill
Client			C R U D	R
Service Engineer			R U E	R
ESM System			U	
Service Order	R	R U		C R U E
Bill	R		R	

Data Management Layer Design

Package Diagram



Database

Relational Database

Client

Service request #	First Name	Last Name	Phone #	Address
00041562	Johhny	Coleman	726 - 497 7355	126 Railroad Avenue, CA 95512
00041563	Peter	Cooper	319 - 173 9094	262 Sycamore Street, CA 90205
00041564	Todd	Harris	888 - 777 5488	43 Fawl Lane, CA 92074
00041565	Jimmy	Allen	776 - 674 3762	504 Garfield Avenue, CA 94806
00041566	Diane	Morgan	365 - 836 8665	190 Church Street, CA 94006
00041567	Helen	Hughes	779 - 760 7778	693 Old York Rd, CA 95039
00041568	Jennifer	Smith	253 - 496 8955	904 Virginia St, CA 95635
00041569	Cheryl	Bryant	190 - 380 2720	728 B Street, CA 90052
00041570	Gregory	Lopez	685 - 594 5436	786 Lexington Court, CA 93909
00041571	Jerry	Walker	142 - 861 4964	451 Altantic Avenue, CA 91752

Service Order

Service request #	First Name	Last Name	Service Type	Date & Time
00041562	Johhny	Coleman	Bioanalysis	5/14/19
00041563	Peter	Cooper	Washer Repair	5/15/19
00041564	Todd	Harris	Spectrometry	5/15/19
00041565	Jimmy	Allen	Bioanalysis	5/16/19
00041566	Diane	Morgan	SureCycler Thermal Cycling	5/16/19
00041567	Helen	Hughes	Spectrometry	5/16/19
00041568	Jennifer	Smith	SureFish Probing	5/17/19
00041569	Cheryl	Bryant	Spectrometry	5/17/19
00041570	Gregory	Lopez	Bioanalysis	5/17/19
00041571	Jerry	Walker	Microarrays	5/17/19

Bill

Service request #	Amount Paid	Employee #	Address
00041562	$ 83.00	95941625	585 Railroad Avenue, CA 91850
00041563	$ 53.00	34349135	155 Sycamore Street, CA 93447
00041564	$ 97.00	95941625	160 Fawl Lane, CA 91933
00041565	$ 94.00	90978130	426 Garfield Avenue, CA 92850
00041566	$ 104.00	35227952	865 Church Street, CA 91220
00041567	$ 123.00	74415754	897 Old York Rd, CA 90302
00041568	$ 47.00	34349135	69 Virginia St, CA 92005
00041569	$ 50.00	74415754	364 B Street, CA 91586
00041570	$ 92.00	34349135	648 Lexington Court, CA 94667
00041571	$ 54.00	35227952	790 Altantic Avenue, CA 92302

Service Engineer

Employee Name	Employee #	Phone #
Teresa Simmons	38329222	604 - 161 3975
Anthony Scott	92304759	627 - 820 2199
Susan Morris	98505029	582 - 422 9507
Alice Edwards	83535351	165 - 207 7039
Gary Bennett	31073989	459 - 950 8843

MySQL Schema

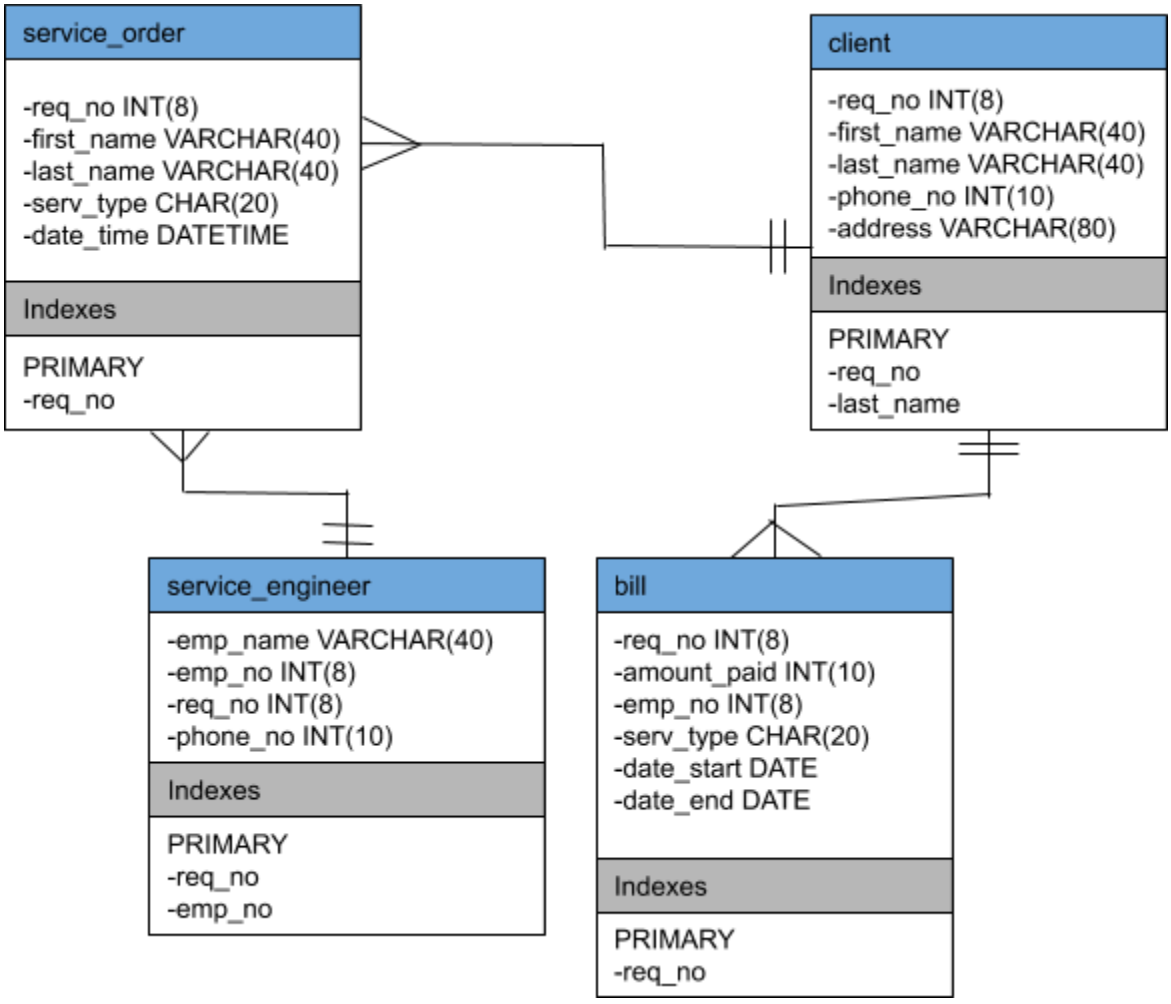
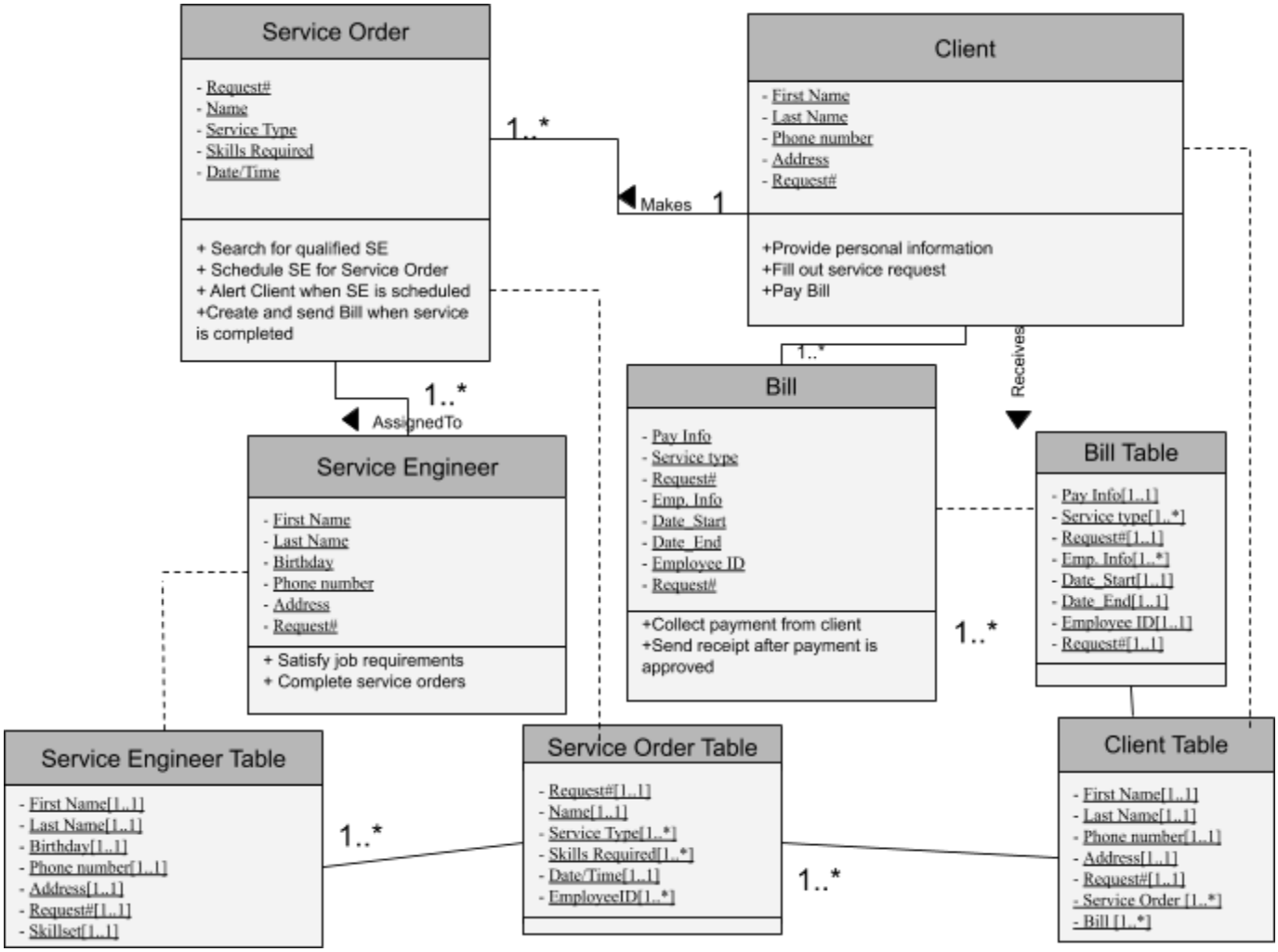


Table Mapping



Human Computer Interaction Layer Design

Use Scenarios

Use Scenario 1: Client Requests a Service

1. Client logs into account
2. If New Service Order
   Client provides information asked on System
3. System displays wait times and availability list
4. Client selects preferred date and time for service
5. Client requests service at said date and time
6. If service if available
   System displays service appointment information
   Else
   System repeats steps 4-6
7. Client confirms appointment and address
8. If Client needs to change information
   Update Client Account
9. System displays confirmation message
10. System creates a Service Order
11. Client receives order number

Use Scenario 2: Search for suitable Service Engineer for Service Order

1. Service Order passes information to Employee Skills Information system
2. System waits for Client confirmation
3. System sends information to Employee Skills Information System
4. If Employee Skills Information System returns a Service Engineer
   System displays slots available
   Client selects best time
   System alerts Service Engineer
   Else
   System displays best matching available Service Engineer slots
   Client selects best appointment or Client changes availability
   System records changes made by Client
   Go back to beginning of step 4

Use Scenario 3: Alert when Service Engineer is Assigned

1. Client confirms Service Order with assigned date and time
2. System updates Service Order with Service Engineer assigned
3. System alerts Service Engineer of new placement

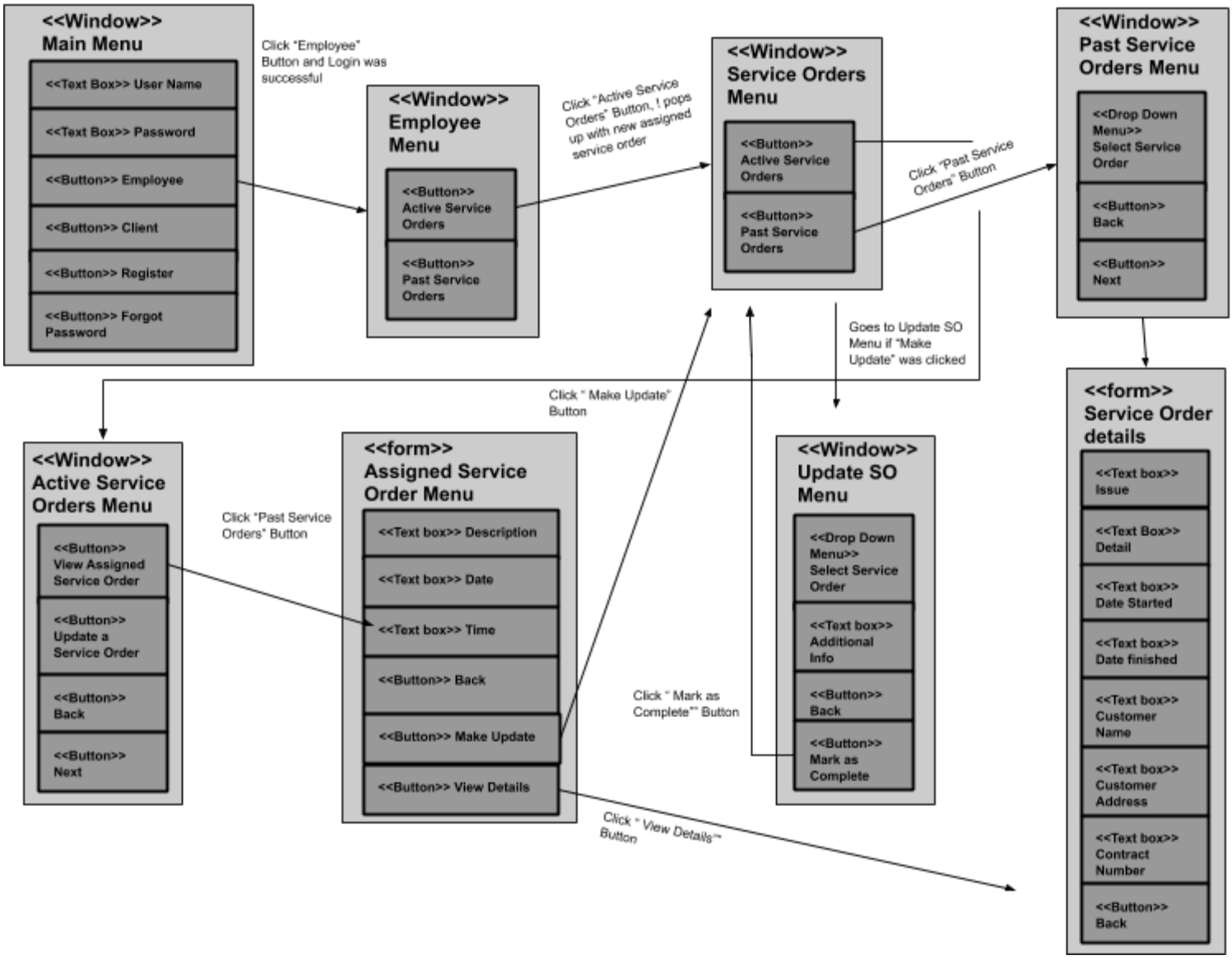
Use Scenario 4: Record when a Service Order has finished

1. Service Engineer logs into the system
2. Inputs Order Number and other necessary information
3. Service Engineer updated Service Order as complete
4. Client rates or provides feedback of Service Order

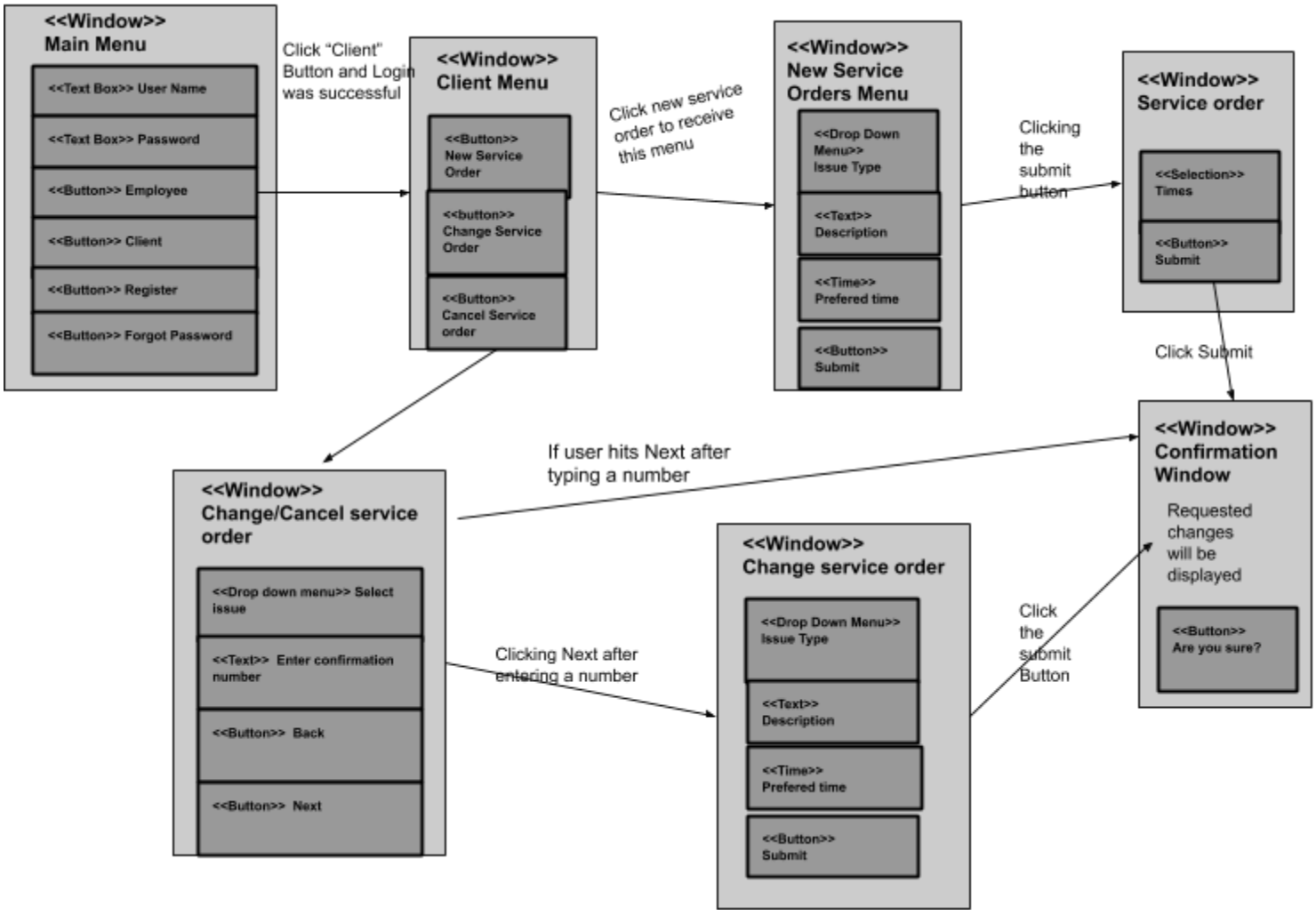
Use Scenario 5: Send the Bill to the Client

1. System processes information on the Service Order
2. System calculates costs and hours worked on order
3. System created a Bill with total amount
4. Bill is sent to Client
5. System displays options to pay bill
6. Client pays bill
7. Client received a receipt of Bill
8. Service Order is archived

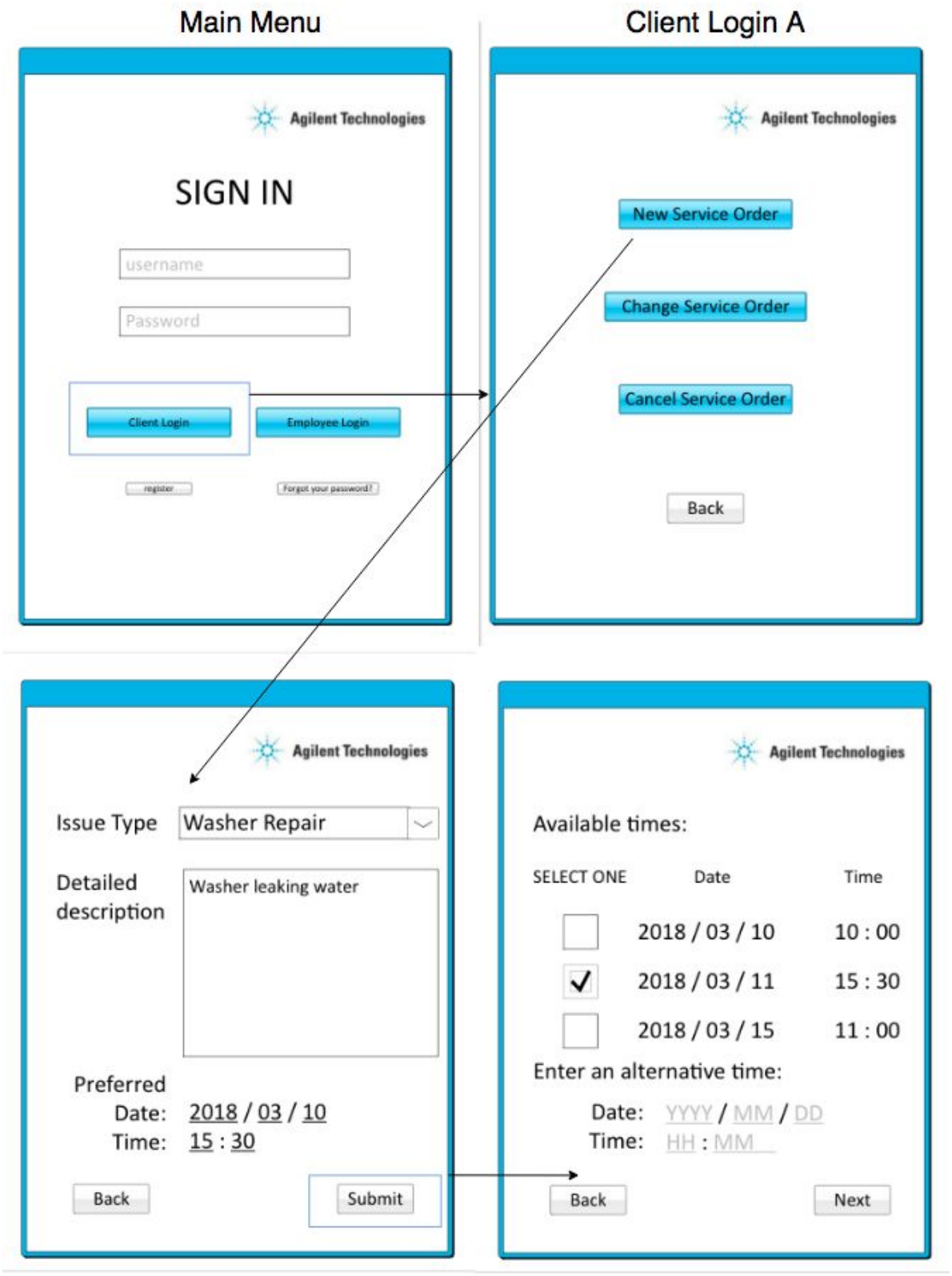
Windows Navigation Diagram



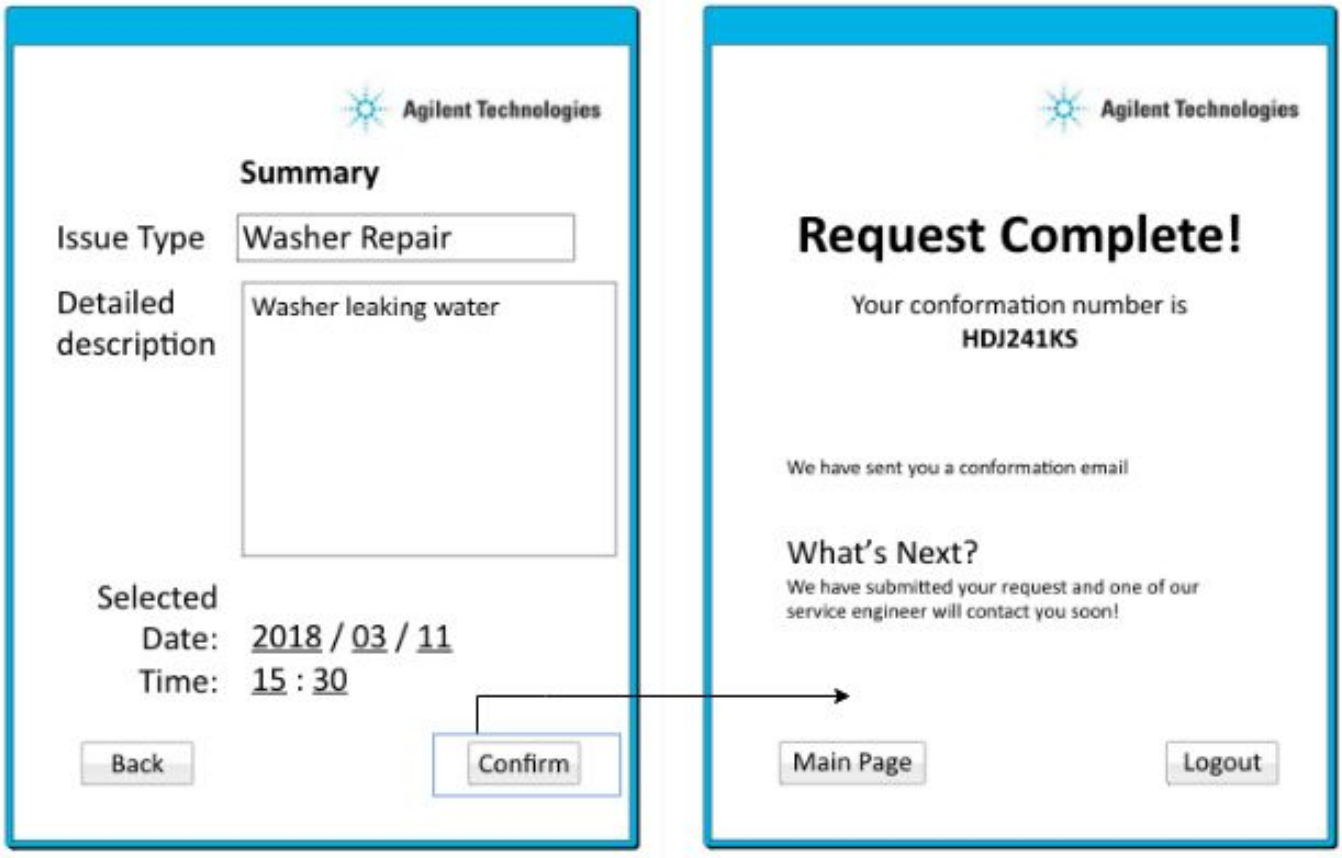
Client Side

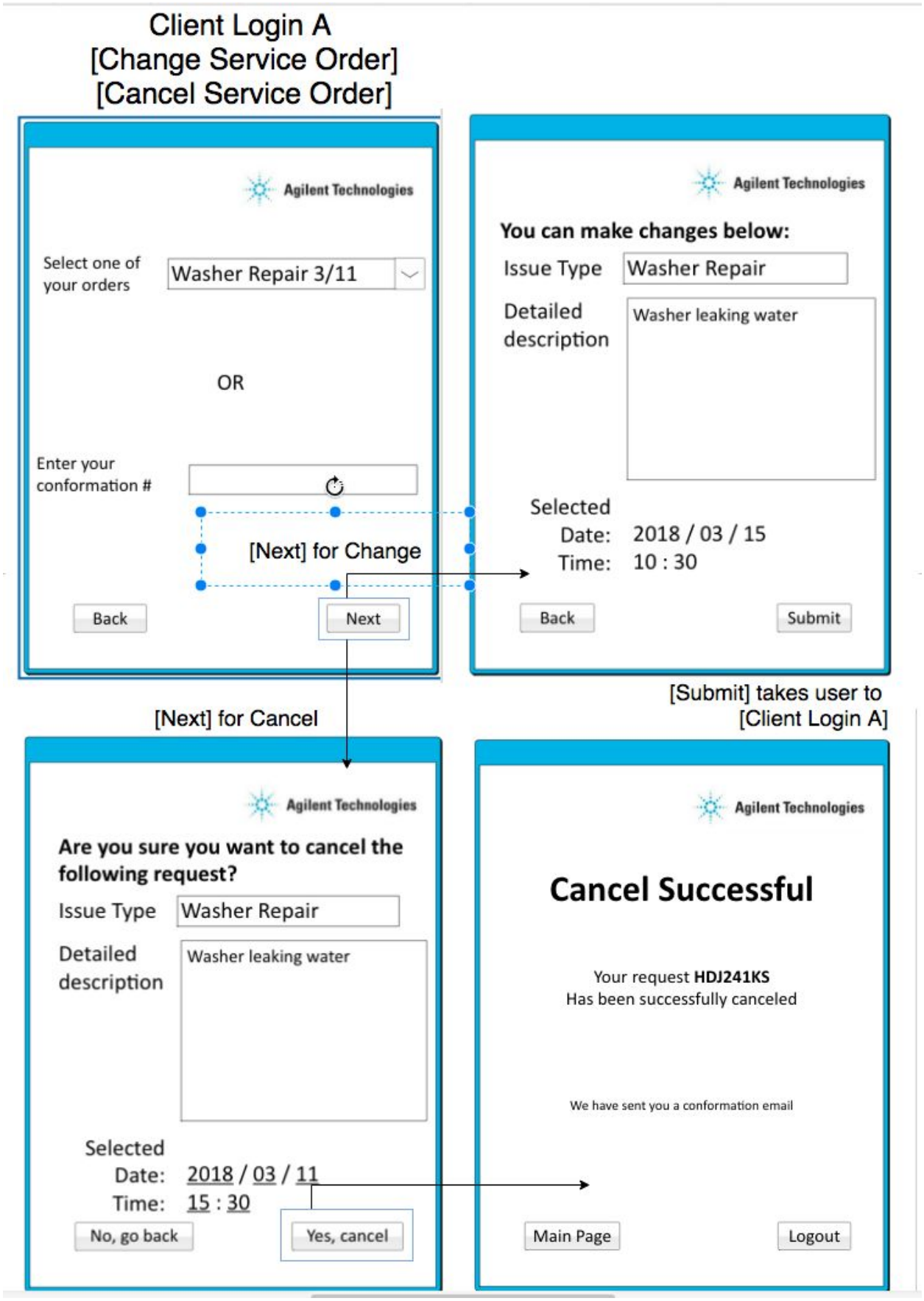

<<Window>>
Main Menu
<<Text Box>> User Name
<<Text Box>> Password
<<Button>> Employee
<<Button>> Client
<<Button>> Register
<<Button>> Forgot Password
Click "Client" Button and Login was successful
<<Window>>
Client Menu
<<Button>> New Service Order
<<button>> Change Service Order
<<Button>> Cancel Service order
Click new service order to receive this menu
<<Window>>
New Service Orders Menu
<<Drop Down Menu>> Issue Type
<<Text>> Description
<<Time>> Prefered time
<<Button>> Submit
Clicking the submit button
<<Window>>
Service order
<<Selection>> Times
<<Button>> Submit
Click Submit
<<Window>>
Confirmation Window
Requested changes will be displayed
<<Button>> Are you sure?
If user hits Next after typing a number
<<Window>>
Change/Cancel service order
<<Drop down menu>> Select issue
<<Text>> Enter confirmation number
<<Button>> Back
<<Button>> Next
Clicking Next after entering a number
<<Window>>
Change service order
<<Drop Down Menu>> Issue Type
<<Text>> Description
<<Time>> Prefered time
<<Button>> Submit
Click the submit Button

Windows Layout Diagram



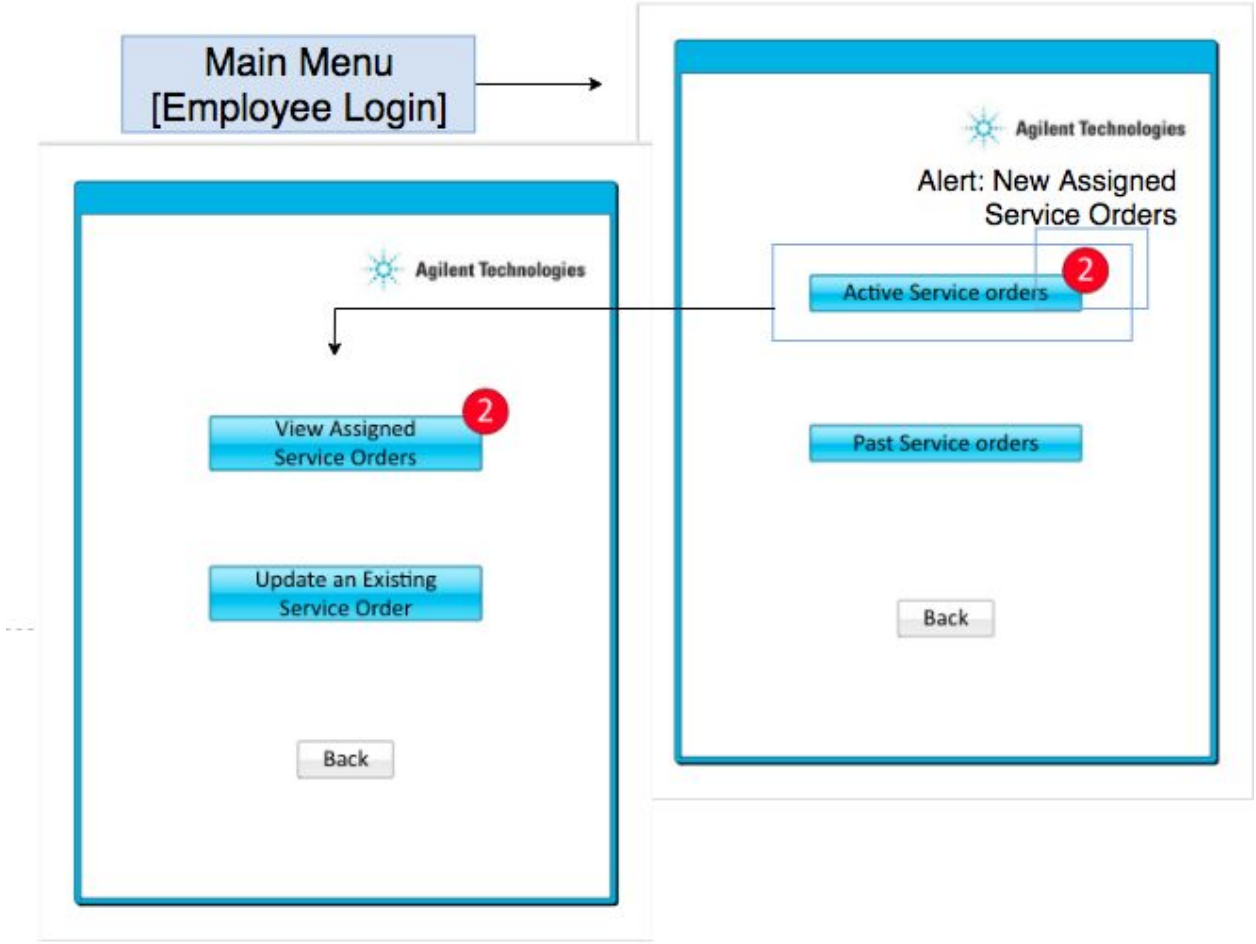
Client Continued. (using Washer as arbitrary example)

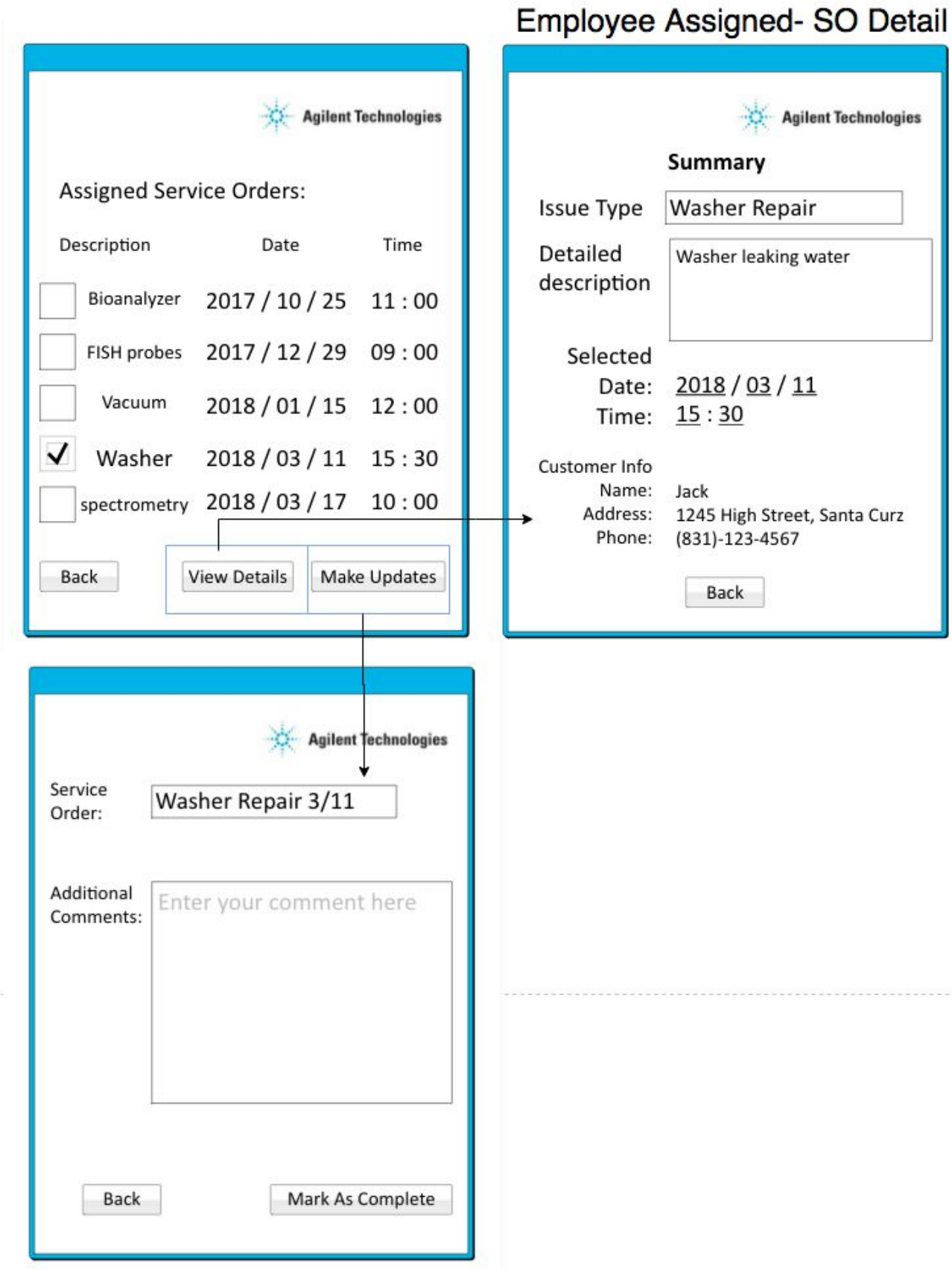


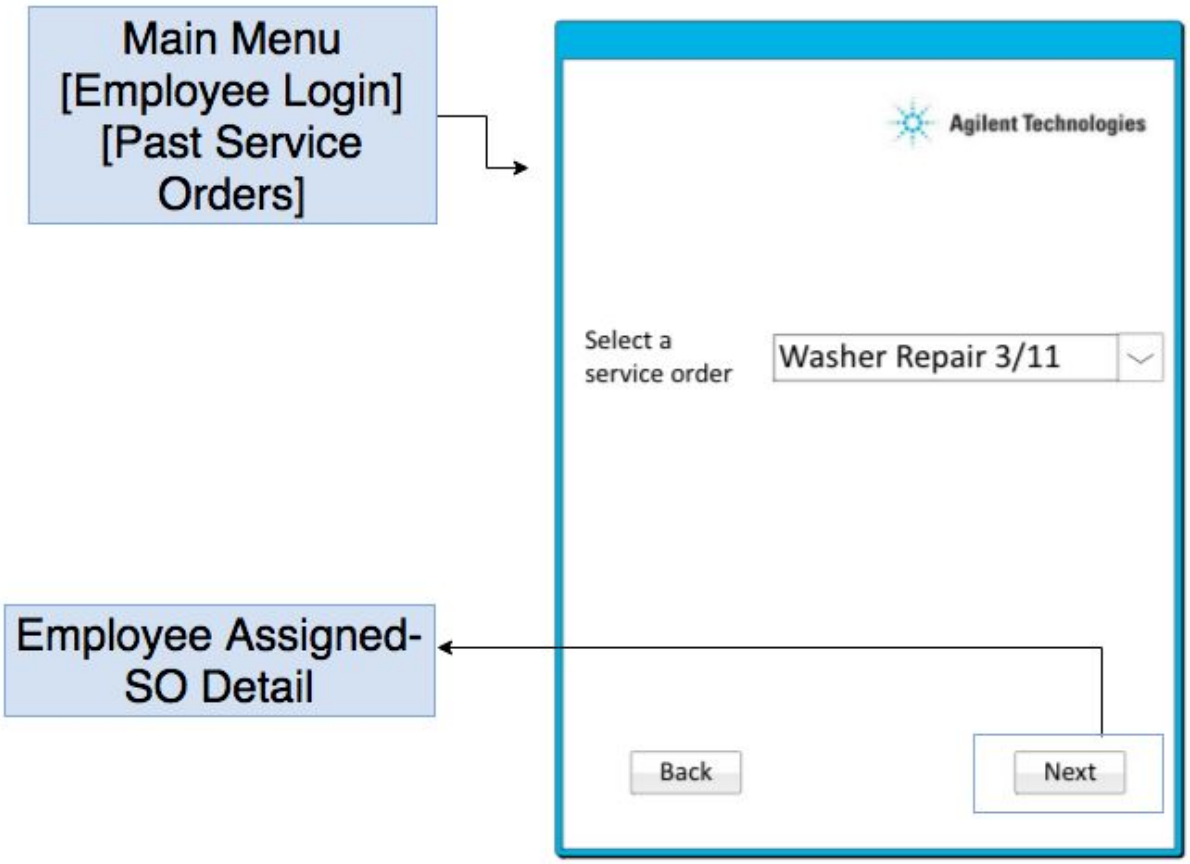

Client Login A
[Change Service Order]
[Cancel Service Order]
Agilent Technologies
Select one of your orders
Washer Repair 3/11
OR
Enter your conformation #
[Next] for Change
Back
Next
Agilent Technologies
You can make changes below:
Issue Type
Washer Repair
Detailed description
Washer leaking water
Selected
Date: 2018 / 03 / 15
Time: 10 : 30
Back
Submit
[Submit] takes user to [Client Login A]
[Next] for Cancel
Agilent Technologies
Are you sure you want to cancel the following request?
Issue Type
Washer Repair
Detailed description
Washer leaking water
Selected
Date: 2018 / 03 / 11
Time: 15 : 30
No, go back
Yes, cancel
Agilent Technologies
Cancel Successful
Your request HDJ241KS
Has been successfully canceled
We have sent you a conformation email
Main Page
Logout

Employee Login

(Possible Service Order types gathered from Agilent Technologies site as arbitrary examples)




Employee Assigned- SO Detail
Agilent Technologies
Assigned Service Orders:
Description
Date
Time
Bioanalyzer 2017 / 10 / 25 11 : 00
FISH probes 2017 / 12 / 29 09 : 00
Vacuum 2018 / 01 / 15 12 : 00
Washer 2018 / 03 / 11 15 : 30
spectrometry 2018 / 03 / 17 10 : 00
Back
View Details
Make Updates
Agilent Technologies
Summary
Issue Type
Washer Repair
Detailed description
Washer leaking water
Selected
Date: 2018 / 03 / 11
Time: 15 : 30
Customer Info
Name: Jack
Address: 1245 High Street, Santa Curz
Phone: (831)-123-4567
Back
Agilent Technologies
Service Order:
Washer Repair 3/11
Additional Comments:
Enter your comment here
Back
Mark As Complete


Main Menu
[Employee Login]
[Past Service Orders]
Agilent Technologies
Select a service order
Washer Repair 3/11
Employee Assigned-SO Detail
Back
Next

Navigation Design Documentation

Use Case Name: Request a Service	**ID** – #00001	**Importance Level**: High
Primary Actor: Client		**Use Case Type**: Detailed, Real
Stakeholders and Interests: Client - wants to request a service		
Brief Description: Client wants to request a service, and system creates a Service Order		
Trigger: When the client first requests an order **Type**: External		
Relationships: Association: Client Include: Extend: Generalization:		
Normal Flow of Events 1. Client logs into account 2. System asks Client if they want to create, change, or cancel SO If Client wants to create an SO, they click on the New SO link and execute S1: New SO If Client wants to change an SO, they click on the change SO link and execute S2: Change SO If Client wants to cancel SO, they click on the cancel SO link and execute S3: Cancel SO		
SubFlows: S-1: New SO 1. Client fills out form with necessary information 2. System displays wait times and availability list 3. Client selects preferred date and time for service 4. Client requests service at said date and time 5. If service if available System displays service appointment information Else System repeats steps 4-6 6. Client confirms appointment and address S-2: Change SO 1. System lists all Client's active SO		

2. Client chooses one SO
3. System displays information of the SO
4. Client changes information
5. Client confirms changes and submits to System

S-3: Cancel SO

1. System lists all Client's active SO
2. Client chooses one SO
3. System displays information and asks for Client's confirmation of deletion
4. If Client clicks the Back link

   System doesn't change anything

   Else

   System removes SO

Alternate/Exceptional Flows:

S-1a, S-2a, S-3a: Connection error and changes aren't saved

1. System produces an error message

Use Case Name: Search for a suitable Service Engineer for new Service Order	**ID** – #00002	**Importance Level**: High
Primary Actor: Service Order		**Use Case Type**: Detailed, Real
Stakeholders and Interests: Client - Needs a service engineer to fulfill their request Service Order - Searches for Service Engineer by matching with Employee Skills Information		
Brief Description: This use case describes how to find a Service Engineer for the Service Order		
Trigger: Request file for new service engineer **Type**: Internal		
Relationships: Association: Client, Service Engineer, Employee Skills Info Include: Extend: Generalization:		
Normal Flow of Events 1. Service Order passes information to Employee Skills Information system 2. System waits for Client confirmation		

3. System sends information to Employee Skills Information System
4. If Employee Skills Information System returns a Service Engineer
   Execute S-1: SE Found
   Else
   Execute S-2: SE Not Found

SubFlows:

S-1: SE Found

1. System displays slots available
2. Client selects best time
3. System alerts Service Engineer

S-2: SE Not Found

1. System displays best matching available Service Engineer slots
2. Client selects best appointment or Client changes availability
3. System records changes made by Client
4. Go back to step 4 in normal flow of events

Alternate/Exceptional Flows:

S-2a: SE not found because none are available

1. produces an error message

A-2: Connection error and changes aren't saved

1. System produces an error message

Use Case Name: Alert that Service Engineer is assigned	**ID** – #00003	**Importance Level**: High
Primary Actor: Client		**Use Case Type**: Detailed, Real
Stakeholders and Interests: Clients - confirms appointment Service Order - adds Service Engineer Service Engineer - notified of placement		
Brief Description: This use case describes what happens when an Service Engineer is assigned		
Trigger: Client has selected Service Engineer slot **Type**: External		
Relationships: Association: Client, Service Engineer Include: Extend: Generalization:		
Normal Flow of Events 1. Client confirms Service Order with assigned date and time 2. System updates Service Order with Service Engineer assigned 3. System alerts Service Engineer of new placement		
SubFlows:		
Alternate/Exceptional Flows: A-1: Connection error and changes aren't saved 1. System produces an error message		

Use Case Name: Record when the SO has been serviced	**ID** – #00004	**Importance Level** – High
Primary actor: Client, Service Order		**Use case type**: Detail, Real
Trigger: When the Client has been serviced the client rep will notify the system and we will begin processing the receipt that is emailed to them.		
Type: External		
Relationships Association: Client, Service Engineer Include Extend Generalization		
Normal flow of events: 1. Service Engineer logs into the system 2. Service Engineer clicks on View Assigned SO link 3. Service Engineer clicks on Update button 4. Service Engineer inputs necessary search information 5. Service Engineer updates Service Order as complete 6. Service Engineer clicks on Save button 7. Client rates or provides feedback of Service Order		
Subflow:		
Alternate/Exceptional Flow: A-1: Connection error and changes aren't saved 1. System produces an error message		

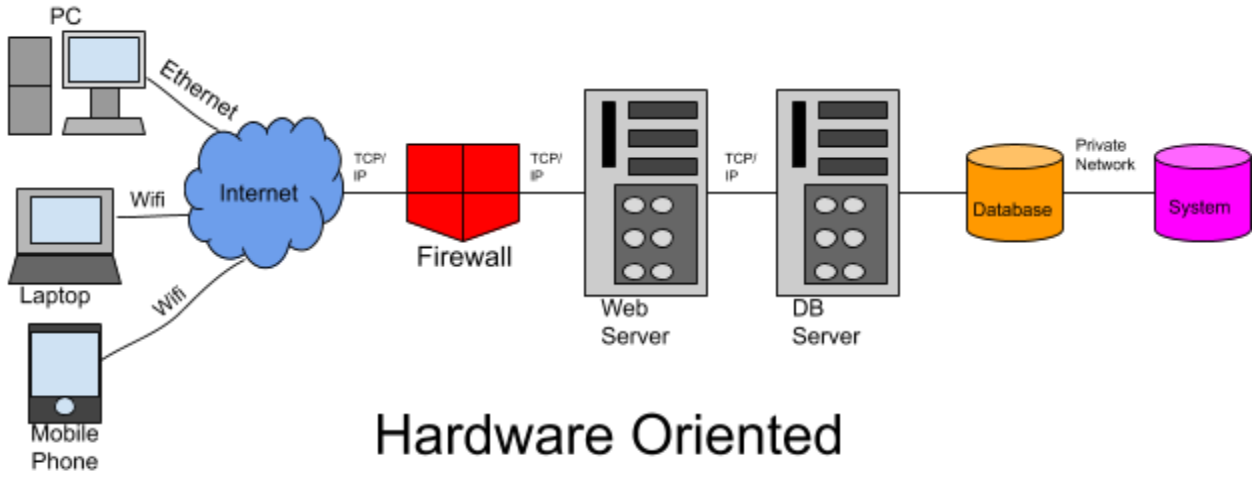
Use Case Name: Send Bill to Client	**ID** – #00005	**Importance Level** – High
Primary actor: Client, Service Order, Bill		**Use case type**: Detail, Real
Trigger: Process the Service Order and bill Client for services		
Type: External		
Relationships Association: Client, Service Order Include: Extend: Generalization:		
Normal flow of events: 1. System processes information on the Service Order 2. System calculates costs and hours worked on order 3. System creates a Bill with total amount 4. Bill is sent to Client 5. System displays options to pay bill 6. Client pays bill 7. Client received a receipt of Bill 8. Service Order is archived		
Subflow:		
Alternate/Exceptional Flow: A-1: Connection error and changes aren't saved 1. System produces an error message		

Physical Architecture Layer Design

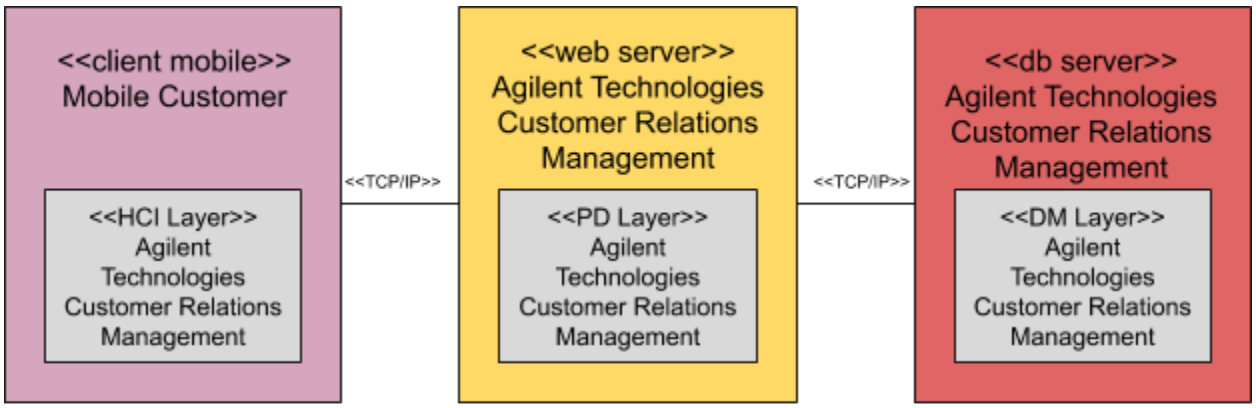
Hardware and Software Specifications

	Client	**Web Server**	**Application Server**	**Database Server**
Operating System	- Windows - Any web browser	- Linux	- Linux	- Linux
Special Software	- Adobe Reader - Adobe Flash	- Apache	- Java	- MySQL
Hardware	- 8 GB RAM - 8 GB Memory - 512 GB disk drive - Intel Core i7	- 256 GB RAM - 1 TB disk drive - Intel Xenon	- 256 GB RAM - two 1 TB disk drive - Intel Xenon	- 8 GB RAM - 4 1 TB disk drive - Intel Xenon
Network	- 100 Mbps Ethernet - high speed wireless	- 100 Mbps Ethernet	- 100 Mbps Ethernet	- 100 Mbps Ethernet

Deployment Diagram



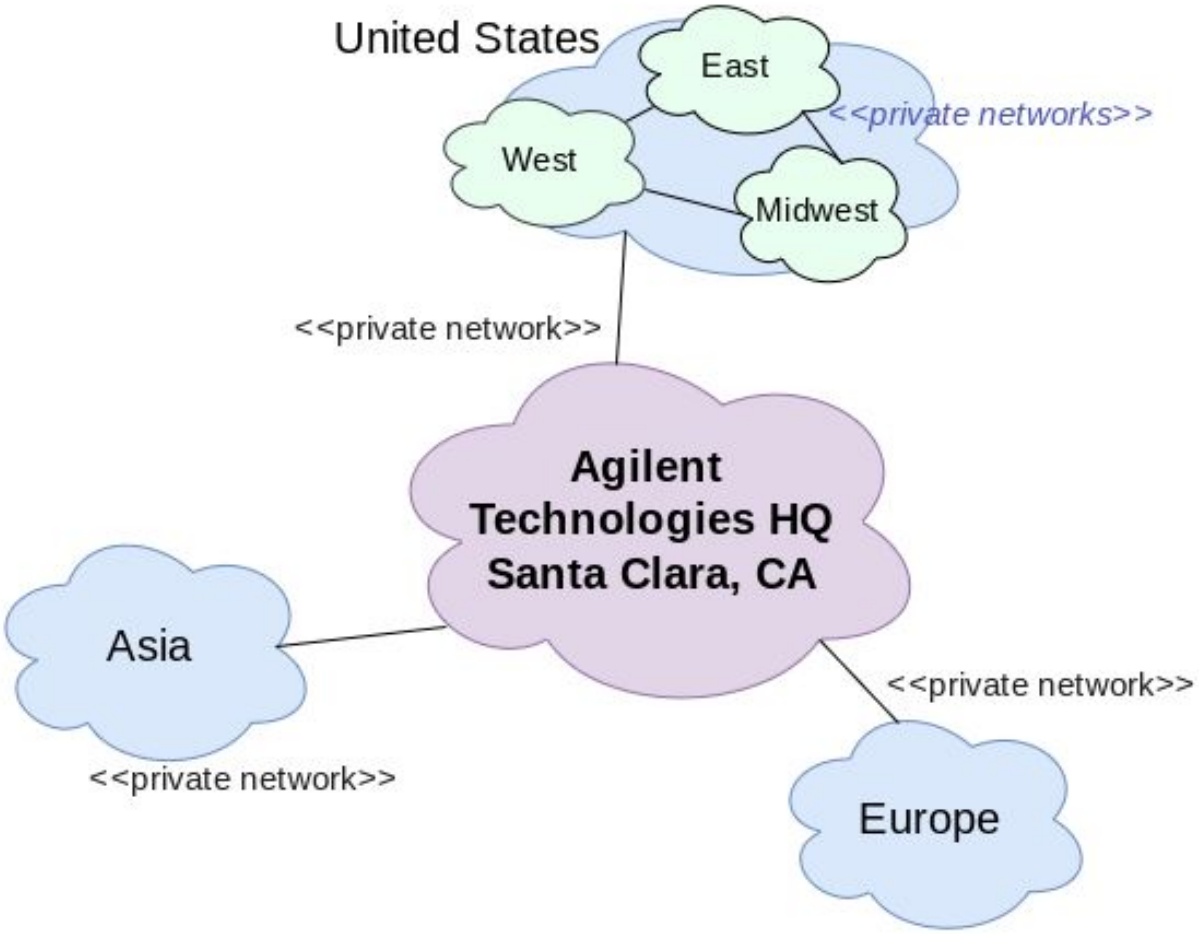
Hardware Oriented Deployment Diagram



Software Oriented Deployment Diagram

Network Model

Higher Level Network Model



Low Level Network Model