Product Mission Statement

Overview

Vending machines took an enormous leap forward in the 1990s when they began to accept credit/debit cards. In the 2010s they took another leap forward when they began to accept other forms of electronic payment. To date, however, this kind of technology has not found its way into the home (or other "home-like" environments). KIkey is about to change that.

KIkey is a system that limits access to food based on a calorie budget, monetary budget, or other kind of budget. KIkey is a hardware/software system consisting of: dispensing devices (and the software that runs on them) that control access to food products, server-side software that manages access, and client-side software (e.g., on mobile devices) that manages "keys".

The mission of KIKey is to:

- Enable you to create and manage a food budget (based on calories, price, or some other measure).
- Enable you to request a specific food item from a server and deduct the "cost" of that item from your budget.
- Enable you to remove approved food items from a dispenser using your cellular phone (or other mobile computing/communication device).

Target Market

We anticipate that KIkey will initially be of interest to a niche market of families that want to control their food intake, primarily for health reasons. Ultimately, we think that KIkey will become a mass market product for home use, and may also be marketable to commercial entities (e.g., hotels).

Our preliminary research indicates that there are no other companies discussing or developing comparable products.

Needs Statement

The KIkey product design team completed a competitive product analysis (over the course of the past two months) and then used two focus groups to arrive at this needs statement. One of the focus groups consisted of existing customers and one consisted of potential customers.

System Needs

S-1 Run on a server in "the kitchen".

S-2 Run on cellular telephones.

S-3 Run on dispensers.

User Needs

A user of KIkey needs to be able to:

U-1 Request a voucher from the server.

U-2 Receive a voucher from the server only if her/his budget (which might be measured in monetary units, calories, or other units) allows.

U-3 Transmit a voucher to a dispenser.

U-4 Receive a food item from a dispenser.

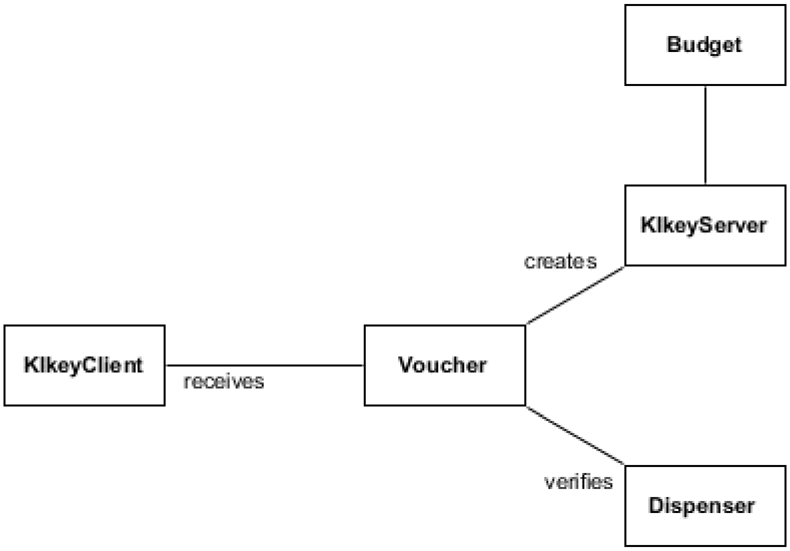
Requirements Specification

Abbreviations and Acronyms

RMI	Remote Method Invocation

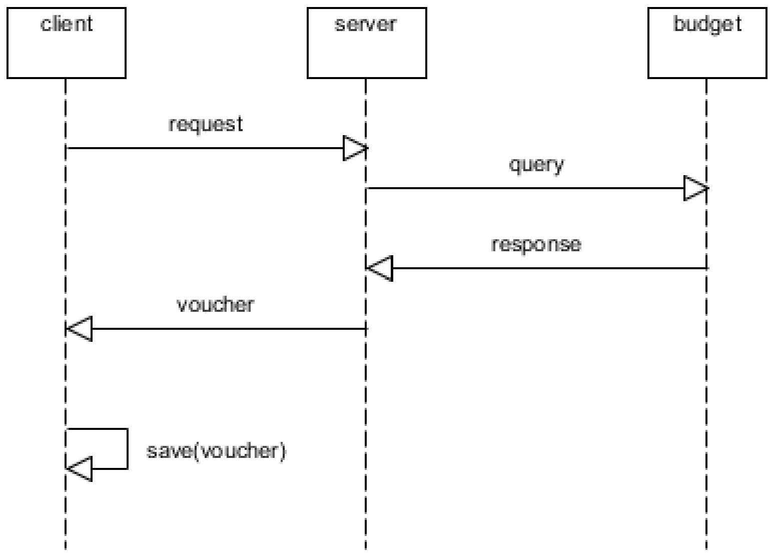
Conceptual Model

The following conceptual model was developed early in the design process:

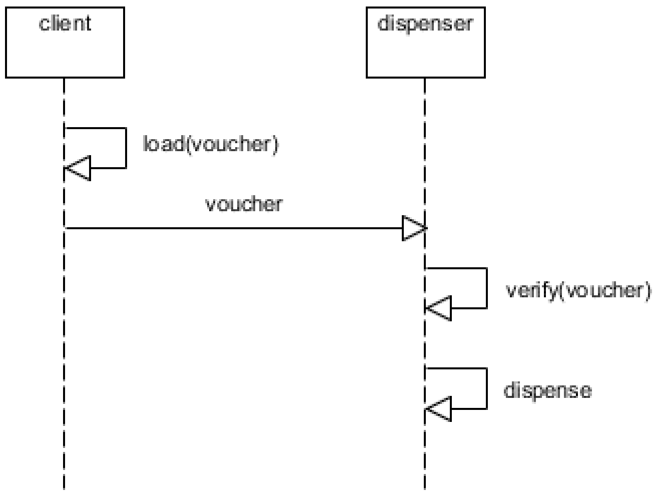


It illustrates, at a fairly high level, the important concepts in the system (note that the WWW browser is not part of the system). It is not intended as an engineering design. That is, the software components that need to be designed and developed may or may not correspond to these concepts.

In lieu of use-case descriptions, the design team created two sequence diagrams that illustrate how the components of the system interact. The first illustrates a user retrieving a voucher.



The second illustrates a user using a voucher to request that an item be dispensed.



Engineering Design Requirements

The system must support multiple communications technologies. Hence, the system must satisfy the following engineering design requirements:

ER-1 The software components that comprise the server must use a layered architecture that distinguishes the components responsible for the “business logic” from the components responsible for the “communications logic”.

ER-2 The software components that comprise the dispenser must use a layered architecture that distinguishes the components responsible for the “business logic” from the components responsible for the “communications logic”.

ER-3 The software components that comprise the client must use a layered architecture that distinguishes the components responsible for the “business logic” from the components responsible for the “communications logic”.

In other words, it must be easy to swap one communications technology for another and code must not be duplicated across communications technologies. The current implementation must be able to work with both Java’s RMI (JavaRMI) technology and a custom SOAP technology.

Operational Requirements of All Systems

All systems (regardless of the communications technology being employed) must satisfy the following operational requirements:

OR-1 The voucher server must be able to search a "budget" to determine whether the user can have access to the requested item. (see Needs U-1)

OR-2 The server must be able to verify that the user's budget (whether measured in calories, money, or other units) is large enough to satisfy the request. (see Needs U-2)

OR-3 The client must be able to store a "voucher" on the file system (i.e., "voucher" objects must be persistent).

OR-4 The client must be able to retrieve a "voucher" from the file system.

OR-5 The dispenser must be able to validate a "voucher".

OR-6 The dispenser must be able to dispense the requested food item when it receives a valid "voucher". (see Needs U-4)

OR-7 Users must not be able to modify vouchers.

OR-8 Vouchers must be able to be used only once.

Physical Requirements of the JavaRMI-Based System

The system must satisfy the following physical requirements:

PR-1 The system must have an RMI-based server that runs "in the kitchen".

PR-2 The system must have an RMI-based component that runs on dispenser devices.

PR-3 The system must have an RMI-based client that interacts with the voucher server and the dispenser. (This client will ultimately run on cellular telephones. For now, this client should also run on a desktop computer.)

Operational Requirements of the JavaRMI-Based System

The system must satisfy the following operational requirements:

OR-9 The voucher server must be able to transmit an electronic "voucher" using RMI. (see Needs U-1)

OR-10 The client must be able to request a "voucher" using RMI. (see Needs U-1)

OR-11 The client must be able to transmit a "voucher" to a dispenser using RMI. (see Needs U-3)

Physical Requirements of the SOAP-Based Components

The system must satisfy the following physical requirements:

PR-4 The system must have a SOAP-based server that runs "in the kitchen".

PR-5 The system must have a SOAP-based component that runs on dispensers devices.

PR-6 The system must have a SOAP-based client that interacts with the voucher server and the dispenser. (This client will ultimately run on cellular telephones. For now, this client should run on a desktop computer.)

Operational Requirements of the SOAP-Based Components

In addition to the relevant operational requirements for the RMI-based components, the SOAP-based components must satisfy the following operational requirements:

OR-12 The voucher server must be able to transmit an electronic "voucher" using SOAP. (see Needs U-1)

OR-13 The client must be able to request a "voucher" using SOAP. (see Needs U-1)

OR-14 The client must be able to transmit a "voucher" to a dispenser using SOAP. (see Needs U-3)