



IntelliToast Requirements Analysis

Version: 1.0

Author: Chris Nippard

Date: 07.17.2022

Contents

1.0 – Scope	3
2.0 – Assumptions.....	3
3.0 – Functional Requirements	5
3.1 – Mandatory Requirements.....	5
3.1.1 – Backend.....	5
3.1.2 – API	5
3.1.3 – Android UI	6
3.2 – Added Value Requirements	6
3.2.1 – Defrost.....	6
3.2.2 – Slice Detector	7
3.2.3 – Notifications	7
3.3 – Exploratory Requirements	7
3.3.1 – Smoke Detector.....	8
3.3.2 – Self Clean.....	8
3.3.3 – AI (Machine Learning & Data Science).....	8

1.0 – Scope

The scope of this document is to identify the functional requirements of the “IntelliToast” product (developed by Mysa) and to provide detailed value of these requirements relative to the business need.

2.0 – Assumptions

Hardware Assumptions:

- 1) Toaster Apparatus – It is assumed that the toaster hardware itself shall be provisioned with the following (software controlled) feature criteria:
 - a) Firmware – controls toaster settings and performs POST logic.
 - b) Logic Controller – embedded software which handles message pump. Messages are transmitted via an API which channels messages to the toaster.
 - c) 4x Servos motor – provides mechanism per each slice detected to descend or ascend into toasting area.
 - d) 4x optical laser – provides mechanism to detect the presence of slice. It is assumed that if no slice is detected in a slice loading area it will return an error indicative of the same, should the user elect 4 slices and there only be 3 slices physically present (for example).
 - e) Physical button for “begin session” should the user wish to interact with the toaster manually.
 - f) Physical button for “cancel session” should the user wish to cancel the current toasting session.

Software Assumptions:

- 1) Database – It is assumed that during it’s initial version, the DB shall be populated with 2 tables initially, as follows:
 - a) Settings – This table contains all of the toasters user and hardware defined settings. Default settings are available via a config file.
 - b) Profiles – This feature shall not be available in V1.0 (the toaster shall auto-authenticate in v1.0), however the table is present as a function of the logical design of the DB. It is assumed that Settings shall be a foreign key on the Profiles table such that each user has its own unique set of configurations (i.e. perhaps the dad user likes his toast a little crispier where the child user may like their toast less crunchy and lighter).

- 2) Cloud Based API – It is assumed that an API shall be made available to interact with the hardware. The API shall be employed as a web service (possibly a function app in Azure or equivalent in AWS). The API's core functionality shall provide the following layer of communication:
- a) Receives a request object (with json body) from the UI's controller and parses the json data.
 - b) Each user's toaster has a unique ID (let's assume the MAC address of the device) and the API shall connect to making initial handshake.
 - c) Queries the user attempting to login in the profiles table (again, as this feature is deprecated in v1.0 the API will always authenticate with "default user") and validates that the user is associated with the ID of the toaster.
 - d) API updates users settings (v2.0)
 - e) Response is returned to user on UI indicating that the user is authenticated and connected to the toaster.
 - f) UI updates such that user can provide their elections with respect to how the toast is prepared and the user pushes the "Toast!" button, followed by a message asking the user "are you sure you're settings are correct?"
 - g) Request is made to API providing all of the user elected toast preferences (i.e. type of bread – toast, bagel, English muffin, etc, the cook time, timer preferences) and the API processes the requests body.
 - h) API calls toaster and passes parsed data to toasters message pump initiating the toasting session.
 - i) Response is passed back to API and ultimately to the Android device which displays message, "Now Toasting!". A button shall be made available under the message, "CANCEL!". Should the user elect to cancel, another payload shall be passed through the API which hands the kill message to the toaster.

3.0 – Functional Requirements

This section is designated into 3 distinctions, respectively:

- 1) Pri 0 – Mandatory Requirements – These requirements “make or break” the product. Without these features, the product is unable to run in its barest capacity.
- 2) Pri 1 – Added Value Requirements – These requirements outline features that are “nice to have”, adding competitive value to the product, outweighing the functionality of our competitors.
- 3) Pri 2 – Exploratory Requirements – These requirements are encompassed within the domain of Internal Research & Development. In an Agile methodology the efforts put forth into these types of requirements are often referred to as “spike work”.

3.1 – Mandatory Requirements

This section defines all mandatory requirements for the product. As this is a living document, these requirements may become revised as later versions become available.

3.1.1 – Backend

The backend shall be comprised of a DB layer (possibly SQL Server or SQLite) hosted in a cloud services environment such as Azure or AWS.

Business Value:

Provides architectural structure for the product and is the backbone of the authentication components and analytical data collection.

3.1.2 – API

The API layer shall provide communications support between the UI, DB and subsequent messages to the IntelliToast hardware.

Business Value:

Provides an asynchronous abstraction layer between the UI and the hardware and facilitates all communication between the UI, hardware and DB.

It is cloud based, which (assuming minimal downtime) provides a solution which is portable and easily maintained, generating less overhead to the company.

3.1.3 – Android UI

The User Interface layer shall (in v1.0) possess all of the standard functionality a conventional toaster would provide:

- a) Ability to set temperature
- b) Ability to set a timer
- c) Ability to begin toasting
- d) Ability to cancel toasting

Business Value:

This feature is required for any engagement with the hardware. It is a core feature and should reflect an easy, enjoyable user experience.

It is extensible, such that as new features are added, so shall the UI evolve.

3.2 – Added Value Requirements

This section defines all added value requirements for the product. As this is a living document, these requirements may become revised as later versions become available.

3.2.1 – Defrost

The IntelliToast product shall implement a feature which allows the user to defrost their choice of bread, should it be frozen prior to toasting. This feature shall implement a short “defrost period” in which heat is applied at low current for a user selected period and resumes normal toasting activities following the defrost period.

Business Value:

This feature provides an edge over the competition' endeavors. It enhances the experience for the user and saves them time in having to perform the prep work themselves.

3.2.2 – Slice Detector

The intelliToast product shall (in a later version) implement a feature which allows slice detection. A laser is placed on one side of the slice slot and a sensor on the other. When a slice is dropped into the slot it interrupts the laser, flagging this slot as filled.

Business Value:

Should this feature be implemented, it would deprecate the need for selecting the desired # of slices, during the UI workflow.

3.2.3 – Notifications

The intelliToast product shall implement a feature which allows the user to receive notifications on their UI when an event is generated (i.e. when the toasting cycle is complete, when a self-clean cycle is complete, etc). These notifications are user configured and comprise a portion of the settings in the DB (whether the user elects to receive notifications - IsNotificationsAvailable : bool).

Business Value:

This feature adds a convenience mechanism to the user allowing them to know (in real time) the status of their toaster. It provides details of events such as cancelling a session, initiating a session, timers, warnings, and errors.

3.3 – Exploratory Requirements

This section defines all exploratory requirements for the product. As this is a living document, these requirements may become revised as later versions become available.

3.3.1 – Smoke Detector

The intelliToast product shall (in a later version) implement a feature which allows smoke detection built into the toaster such that if the toast begins to burn it shall auto cancel the current toasting session and (if elected by the user) send a notification to the user indicative of the same.

Business Value:

This feature provides a safety/fallback mechanism for the toaster, encouraging customers to err on the side of caution (as opposed to purchasing competitors product) and dramatically reduces the risk of a fire in the household.

3.3.2 – Self Clean

The intelliToast product shall implement a feature which allows the user to self-clean their toaster. It initially checks for the presence of slices in the toaster and reports a message to the user, if present, prompting them to remove the slice(s) from the toaster.

Using a software based relay, the toaster applies very high temperature heat in bursts, allowing the remaining sediment in the toaster to char, but ideally not create a significant amount of smoke. After the self-clean cycle completes, 2 doors open at the bottom of the toaster, dropping the remaining waste into a basin beneath the toaster.

Dependencies:

Feature 3.2.2 must be implemented.

Business Value:

Saves user time in having to clean the toaster and provides an enhanced feature unavailable to most conventional toasters.

3.3.3 – AI (Machine Learning & Data Science)

The intelliToast product shall implement a feature which allows Mysa to collect analytical data about the user's experience, the frequency of use (which in turn collects time stamp data about the session, which can collect details like electricity consumption and generate reports). It allows Mysa an insight as to where faults may occur more frequently in the product. This feature would require adding an "Analytics" table to the DB and monitoring the quantity of bucket data collected.

Business Value:

This feature allows Mysa a window into performance of the product and allows insight into the level of satisfaction of the customer (perhaps adding a 5 star rating system which supplies a notification once a month asking, “how happy are you with your intelliToast experience?!”). This data is collected and stored and can provide valuable, quantifiable analytical data which can be shared with stakeholders, ultimately increasing their confidence in the product.