

# **Wasabi A/B Testing Platform**

**Sai Seetha Ram Nomula**

## **Table of Contents**

<b>1. Abstract</b>	3
<b>2. Introduction</b>	4
2.1 What is A/B Testing?	4
2.2 Importance of A/B Testing	5
2.2.1 Increases engagement with customers	5
2.2.2 Enhances campaign effectiveness and optimizes programs for target audience	5
2.2.3 Enhances marketers' awareness and expertise of audience preferences	5
2.3 Wasabi A/B Testing:	5
2.3.1 Type of Experiments	6
2.3.2 Stakeholders	6
2.3.3 Basic Components of Wasabi	6
<b>3 Use Case diagram</b>	7
3.1 Actors	8
<b>4 Class diagram</b>	9
<b>5 State Chart diagram</b>	13
5.1 Process Flow:	13
<b>6 Integrating Wasabi with the Client system</b>	14
6.1 Create an Experiment	14
6.2 Enter the parameters	14
6.3 Create the bucket	15
6.4 Segmentation	16
6.5 Run the experiment	16
<b>7. Developing Wasabi</b>	17
7.1 Using Wasabi's Code style	17
7.2 Extending Server-side modules	17
7.3 Authentication	18
7.4 Authorization	18
7.5 Real-Time Data Exportation	18
7.5.1 Assignment data	18
7.5.2 Impression/Action data	18
<b>8 Case Studies</b>	19

<b>9. Advantages of Wasabi A/B Testing:</b>	23
<b>10. Disadvantages of A/B testing:</b>	23
<b>11. Code Implementation of Wasabi:</b>	25
<b>12. Suggestions:</b>	27
12.1 Improve the Integration ability of the application:	27
12.2 Scalability:	27
12.3 Interoperability:	27
12.4 Moving to Cloud Servers:	27
12.5 Automating the wasabi:	27
<b>13. References:</b>	28

## **1. Abstract**

In today's world, every now and then, numerous new applications are being released into the market with the target of reaching to as many customers as they can. It is as much important to retain the existing customers and keep growing as it is to develop and launch those new applications. This applies to each and every sector, from E-commerce and banking to web designing and mobile applications. To keep up with the heavy competition in today's market, companies come up with various new features periodically to delight the customers. And these new features can be as subtle as adding new fields or boxes in the web Interface with the aim of making it more user friendly, modifying the design layouts, changing fonts or colors or they can be of larger magnitude such as modifying their design models, bringing new interactive features etc.

Most of the companies until recently used to rely on their gut feeling or intuition that this new feature is going to work and bring them directly into the production without any testing. But now, as the competition is growing, and the companies are increasingly using the Analytics in making decisions, they now want to test them on sample users before implementing those new features or deploying them into production environment to ensure it makes an impact to the customers. And they need a platform to perform the A/B Testing.

**Wasabi** is one such platform that has become widely popular in the recent times. It is now being embraced by many companies of the various domains ranging from health care, banking to web designing and E-commerce. Wasabi is an open source, real-time, enterprise-grade A/B Testing platform developed by Intuit Data Engineering and Analytics(IDEA), and battle-tested in FinTech products, servicing over 120 apps across mobile, web and desktop.

## 2. Introduction

### 2.1 What is A/B Testing?

A/B testing, otherwise called split testing, compares two different versions of the same on a channel. These channels include (however are not restricted to):

- Website
- Social media
- Emails
- Mobile apps
- Landing pages
- Digital advertisements
- Single channels on the above-mentioned channels might include:
  - The homepage of a website.
  - A notification on a smartphone of a new product offer
  - Changing the color of login button of an application.

However, by what amount do the two versions vary? Everything boils down to one component, regardless of whether that is the measure of an interactive button, the wording of a call to act upon, the textual styles used, or another component concerning configuration, design, formation or language. For example, one could set up an a/b test that compares a red color background versus a blue color background for a company website homepage. The page has had a blue color background for quite a long time, yet the company has seen consistent waning conversions furthermore, needs to know whether users are allured by an adjustment in color and therefore might be more responsive to a green color background. Running an A/B test would measure which color background has a more noteworthy impact on click through rates and site visits. If the blue background prompts a significantly higher number of conversions, and the green background drives customers away, the company will know which color will help them meet their business goals.

## **2.2 Importance of A/B Testing**

By surveying the actions of buyers, A/B testing uncovers what genuinely advances to them. In doing as such, it additionally progresses customer engagement, campaign effectiveness, and marketer expertise. Here are a couple of reasons why A/B testing ought to be a piece of your organization's advertising procedure.

### **2.2.1 Increases engagement with customers**

The objective of A/B testing is to progress interactions amongst purchasers and brands by revealing which marketing techniques most reverberate with individuals. Since this sort of testing applies to everything, from more engaging personalized emails to social networks, it opens all channels to stronger communication.

### **2.2.2 Enhances campaign effectiveness and optimizes programs for target audience**

By trying out different combinations for a specific group of customers, marketers can eliminate elements that alienate users, drive people away, or have no effect on conversion rates whatsoever. In addition, all audiences do not respond identically to a single campaign.

### **2.2.3 Enhances marketers' awareness and expertise of audience preferences**

A/B testing provides businesses with enormous amounts of data on audience behaviour. The more tests advertisers run, the more vigorous their understanding will be, what's more, they can start to make more instinctive decisions in their marketing

## **2.3 Wasabi A/B Testing:**

The Wasabi AB Testing service has been developed by Intuit Data Engineering and Analytics (IDEA) based on a set of RESTful APIs that allow us to instrument our product code to perform AB tests (also known as split tests), and then initiate and manage experiments and track and analyze results. In addition to that, a user interface to manage the experiments and to

track results will be provided. This is a simple, flexible, and scalable way to run AB test experiments in your product, you're in the right place.

Wasabi enables the customers to:

- Initiate and manage experiments via API or through its web UI.
- Instrument the customer's product code to execute an AB test experiment via easy REST API calls:
- Assign users to a bucket (A, B, C, ...), then they will be shown the version of the experience the customer's admin wants them to see (treatment).
- Log impressions when users are exposed to the treatment.
- Log a single or multiple type of actions the users may take in response to the treatment (your success metrics).
- Analyze the experiment results to find the successful bucket and verify statistical significance (via API or web UI).
- Immediately roll out the successful experience to all users until the product code is implemented.

### **2.3.1 Type of Experiments**

- Visual elements and content (e.g. red button vs. blue button)
- Multi-page work flows
- Dynamic or single-page client experiences
- Back-end algorithms (e.g. compare recommendation algorithms) etc.

### **2.3.2 Stakeholders**

The Stakeholders list range from the web and mobile clients to the large Analytical organizations that want to perform experimentations on their new features. Few of the stakeholders list include:

- Web clients
- Mobile clients
- Desktop clients (with an internet connection)
- Server-side clients
- Any users or developers from client-side or server-side
- Users on multiple platforms but having a common user identifier

### **2.3.3 Basic Components of Wasabi**

#### **2.3.3.1 Buckets**

Buckets are assigned a set of users randomly for conducting an experiment (bucket A, bucket B, etc.) They represent the experience the users set in them will be offered. Most of the times it is called by other, equivalent names such as variation, treatment group, or recipe.

#### **2.3.3.2 Assignment**

It is a method in which the potential subject of the experiment(users) are assigned to one of the buckets, or decided not to be part of the experiment. Two roll System is used for the assignment: “roll the dice” once and compare to the Sampling percentage to check whether the user is in the experiment or not. If the user is in the experiment, then we roll the dice again to determine which bucket the user is in, depending on the bucket allocation percentages.

These are durable as in once a customer (e.g. UserId) is assigned to any bucket for an experiment, same bucket will always be assigned for the assignment for that UserId, so the users always get similar experience in the product. This also holds even when the user has been decided not to be included the experiment (known as a “null assignment”), which means, when the user’s assignment is called in the Wasabi AB Testing system, “null assignment” will be given.

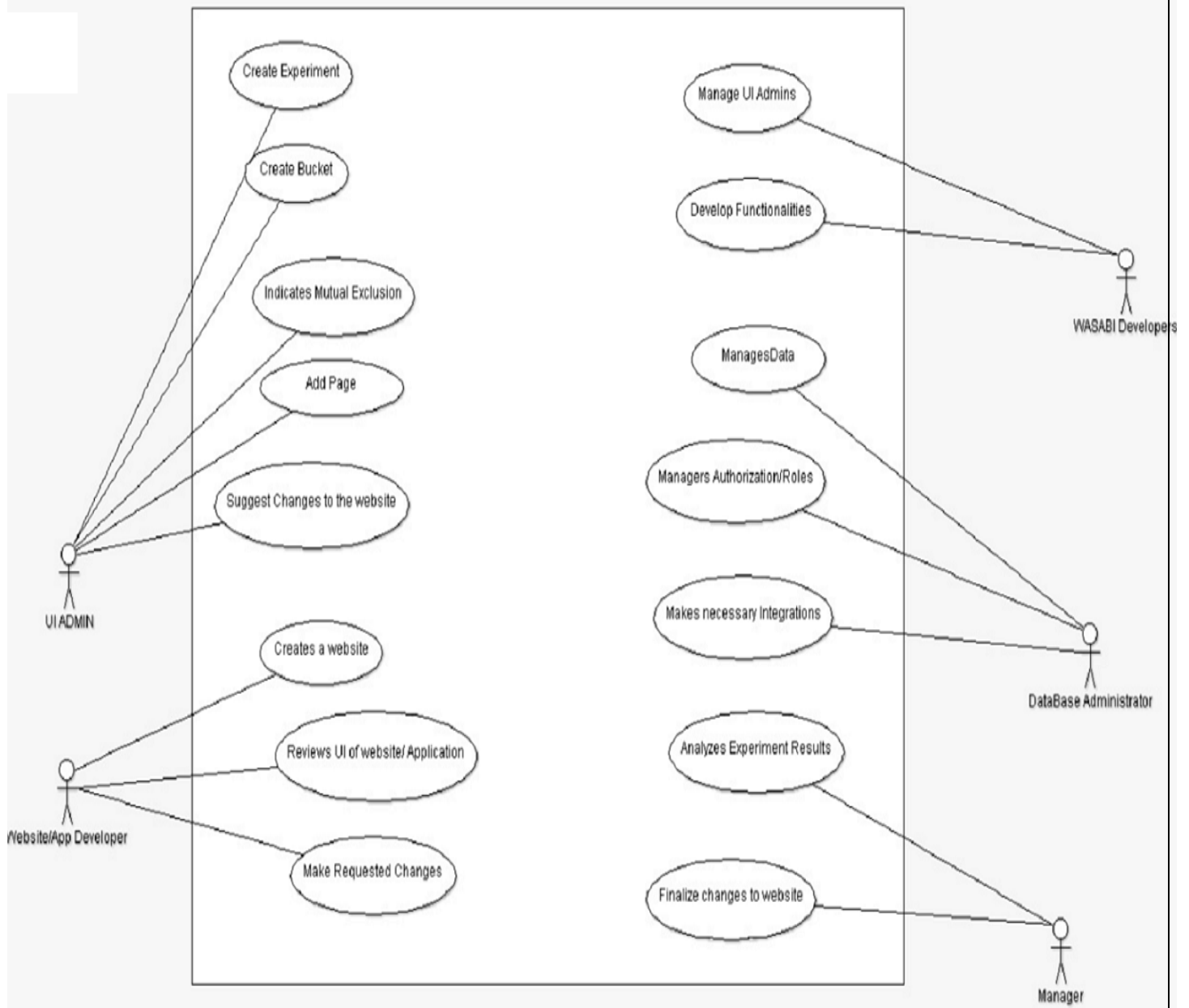
#### **2.3.3.3 Impression**

When a user is allotted to a bucket and is shown the initial part of the corresponding user experience or treatment, an impression will be recorded for that user. It helps in calculating the required statistics. It is logically different from assignment, since, in some situations we may assign a user to a bucket before they get to the part of our product where they see the impression of the user experience.

#### **2.3.3.4 Action**

Subsets of all the users who view an impression of a certain user experience can take a single or multiple action that they are interested in logging and monitoring the metrics to call it a success for that treatment. Every different kind of action corresponds to an action rate (actions divided by impressions, usually confining to unique users). Calculating these action rates and verifying when they differ in a statistically significant way is the object of AB testing.

### **3 Use Case diagram**



### 3.1 Actors

#### UI Admin:

Responsible for setting up the Experiments and personalizing the interface. According to their requirement, he enters all the parameters for the experiments and buckets and suggest changes if any.

#### Web site Developer:



Creates and maintains the website and makes changes to the User Interface or release any new features to the UI.

**Wasabi Developer:**

Manages the UI admin and also responsible for building the functionalities required for performing the A/B Testing.

**Data base Administrator:**

Provides all the authorization roles and make necessary integrations and maintains the data base of records.

**Manager:**

Analyze the Experiment results and finalize the changes made to the website.

## **4 Class diagram**

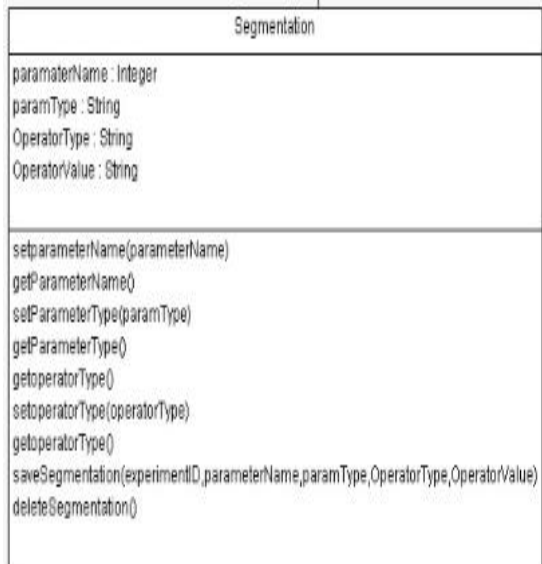
The classes that are essential in this design are as follows:

- Experiments
- Buckets
- Segmentation
- Pages

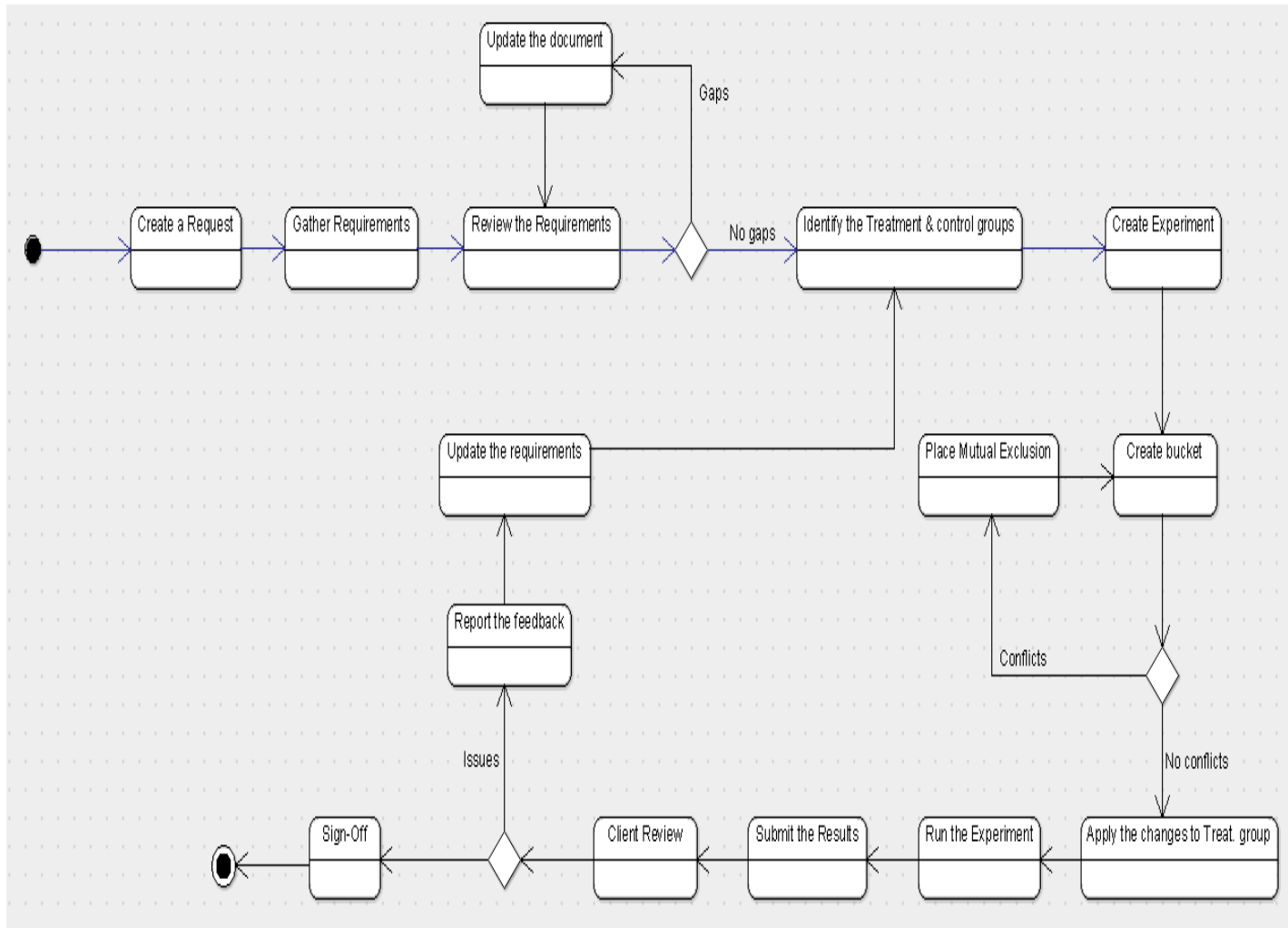
The classes Bucket, Segmentation and Pages are inherited to the Experiment class.

Class Name	Attributes	Operations
<b>Experiment</b>	experimentId: Integer experimentName: String applicationName: String samplingRate: String startDate: DateTime endDate: DateTime rapidExperimentMaxUsers: Integer Description: String	<ul style="list-style-type: none"> <li>• getExperimentId():Integer</li> <li>• setExperimentId(experimentId)</li> <li>• getExperimentName():String</li> <li>• setExperimentName(experimentName)</li> <li>• setApplicationName():String</li> <li>• getApplicationName(applicationName)</li> <li>• setStartDate():DateTime</li> <li>• getStartDate(startDate)</li> <li>• setEndDate():DateTime</li> <li>• getEndDate(endDate)</li> <li>• setRapidExperimentMaxUsers():Integer</li> <li>• getRapidExperimentMaxUsers(rapidExperimentMaxUsers)</li> <li>• getDescription():String</li> <li>• setDescription(description)</li> <li>• fetchExperiment (experimentId, experimentName,applicationName,samplingRate,startDate, endDate,rapidExperimentMaxUSers,description)</li> <li>• downloadAssignment (experimentId, experimentName,applicationName,samplingRate,startDate, endDate,rapidExperimentMaxUSers,description)</li> <li>• addMutualExpt()</li> <li>• cancelMutualExpt()</li> <li>• searchMutualExpt()</li> </ul>
<b>Segmentation</b>	parameterName: String paramType: String operatorType:String operatorValue:String	<ul style="list-style-type: none"> <li>• setParameterName(parameterName)</li> <li>• getparameterName():String</li> <li>• setParamType(paramType)</li> <li>• getParamType():String</li> <li>• setOperatorType(operatorType)</li> <li>• getOperatorType():String</li> <li>• saveSegmentation(experimentId,parameterName, paramType, operatorType,operatorValue)</li> <li>• deleteSegmentation(parameterName)</li> </ul>
<b>Bucket</b>	bucketId:Integer bucketName:String Allocation:String Description:String bucketPayLoad:String Actions: String	<ul style="list-style-type: none"> <li>• getBucketId():Integer</li> <li>• setBucketId(bucketId )</li> <li>• setBucketName(bucketName)</li> <li>• getBucketName():String</li> <li>• getAllocation():String</li> <li>• setAllocation(allocation)</li> <li>• getDescription():String</li> <li>• setDescription(description)</li> <li>• getBucketPayLoad():String</li> <li>• setBucketPayLoad(bucketPayLoad)</li> </ul>

		<ul style="list-style-type: none"><li>• getActions():String</li><li>• setActions(actions)</li><li>• saveBucket(bucketId,bucketName,description,allocation,bucketPayLoad)</li><li>• fetchBucket(bucketId)</li></ul>
<b>pages</b>	pageId : Integer pageName : String	<ul style="list-style-type: none"><li>• getPageId():Integer</li><li>• setPageId(pageId)</li><li>• getPageName():String</li><li>• setPageName(pageName)</li><li>• addPage(pageId,pageName)</li><li>• removePage(pageId)</li><li>• fetchPage(pageId)</li></ul>



## 5 State Chart diagram



### 5.1 Process Flow:

- The Customer makes a Request by logging in to the wasabi system.
- Then, the requirements are gathered and reviewed by the admin. If there are any gaps in the requirements, those are communicated with the customer and the document will be updated.
- Once the user is ready to perform the experiment, first he/she needs to identify the control and treatment groups.
- Then the Experiment should be created, and all its parameters are updated according to the requirement.

- Buckets are to be created and labelled as control and treatment groups. There should be only one control bucket and all the percentage allocations should add up to 100.
- Buckets are checked for Mutual Exclusion and the conflicts, if any, are resolved.
- Then, the Experiment is executed, and the results are submitted to the Client manager.
- Client Manager reviews the files and analyzes the results and takes appropriate steps and the process ends.

## 6 Integrating Wasabi with the Client system

With Wasabi, integrating with any system is as simple and easy as an API call. It can be done in 5 simple steps as follows:

### 6.1 Create an Experiment

Users can Create, Edit, Start, Stop or Terminate Experiments in the Wasabi A/B testing systems. The admin user interface of the system shows a list of the present experiments, arranged by the App and Experiment name.

Experiments

Priority

Welcome, jabba\_writer

Sign out

Feedback

Create Experiment

☒ Hide terminated

+

FAVE	APP ▲	EXPERIMENT	CREATED BY	SAMPLING %	START	END	MODIFIED	STATUS ⓘ	ACTIONS
☆	DEMO	services_portal_doc_test	admin	100	May 23, 2016	Feb 16, 2017	May 23, 2016		
☆	DEMO	bucketTest	admin	100	May 23, 2016	Jun 06, 2016	May 23, 2016		

1 - 2 of 2

Previous

1

Next

### 6.2 Enter the parameters

Here, all the details such as the Experiment name, application name (such as CTG, SBG, CG etc.,) are maintained. Along with that a sampling percentage value is also specified. This indicates what % of the users need to be considered for conducting the experiment. The experiment starts on the given start date and the data is collected and the users are assigned to the buckets. The experiment automatically ends once the given end date is reached.

Create Experiment

Experiment Name ⓘ

Application Name ⓘ

Sampling Rate ⓘ  
 %

Start Date  

 :  AM

End Date  

 :  PM

Rapid Experiment Max Users ⓘ

Experiment Description

Create Experiment

Cancel

### 6.3 Create the bucket

Once the experiment is created, buckets need to be created to assign them to users. A bucket can either be a Control bucket or a treatment bucket. Users assigned to the control bucket will get the default experience or the old features are continued for them. But the users in the treatment group will experience the new features that the client wants to test them on. There should be at least one control bucket. Here, one will also need to fill the Allocation percentage. The allocation percentages for all the buckets should add up to 100.

Create Bucket

Bucket Name

Allocation  
 %
☒ Control group ⓘ

Description

Bucket Payload ⓘ

Add

Cancel

## 6.4 Segmentation

Targeting rules can be given for these experiments. Also, the user admin has an option of choosing which segment of users will see this experiment. A set of users can be targeted to experiment, by defining a certain criterion. For example, if any customer wants to target users whose salary is greater than \$80,000 and who live in California, the segmentation rule would be: salary > 80000 & state = "CA". The customer doesn't need to have any knowledge on any kind of syntax and can easily understand the functions and set the segmentation rules

The screenshot shows the 'Edit Experiment' interface. At the top, there are fields for 'Experiment Name' (services\_portal\_doc\_test), 'Application Name' (DEMO), and 'Sampling Rate' (100 %). Below these are 'Start Date' (4/5/2016 12:00 AM) and 'End Date' (2/24/2017 11:00 PM). A 'Rapid Experiment Max Users' field shows 1-2147483647. A text box contains the message: 'This is a sample experiment created to document the process for the Services Portal.' Below this is a tabbed interface with 'Buckets', 'Mutual Exclusion', 'Segmentation', 'Pages', 'Personalization', and 'API Calls'. The 'Segmentation' tab is active, showing a 'User Segmentation Rule' section. This section contains two rules: 'salary' (number) 'is greater than' '80000' and 'state' (string) 'equals' '"CA"'. There are 'Add rule...', 'Test rule...', and 'Switch to text view...' links. At the bottom right are 'Save' and 'Cancel' buttons.

## 6.5 Run the experiment

Once the experiment and the buckets are created and all the corresponding parameters are maintained, the Play button needs to be clicked in order for the experiment to start. The experiment can be started even before the specified start date by clicking the play button. And the experiment automatically ends on the specified end date. During this time, continuously, the data is collected, and the users are assigned to the buckets.



Experiments		Priority						
Create Experiment		<input checked="" type="checkbox"/> Hide terminated <input type="text" value="Search"/> <a href="#">More...</a>						
APP*	EXPERIMENT	SAMPLING %	START	END	MODIFIED	STATUS	ACTIONS	
app_x	label_x	50	Jun 11, 2014	Jun 12, 2014	Jun 10, 2014			
BucketTestApp	BucketTest	1	Jul 09, 2014	Aug 30, 2014	Jul 11, 2014			
CA_TTO_test	POC_Test	10	Jun 10, 2014	Jun 14, 2014	Jun 10, 2014			
CTG	tto_ty14_B61	3	Aug 07, 2014	Oct 22, 2014	Sep 04, 2014			
DEMO	services_portal_documentation_...	100	Sep 08, 2014	Sep 10, 2014	Sep 09, 2014			
DEMO	ctgqa_demo_Sep82014	100	Sep 08, 2014	Sep 15, 2014	Sep 08, 2014			
Demo_App	MutuaExc	30	Sep 08, 2014	Sep 09, 2014	Sep 08, 2014			
Demo_App	BuyButton	100	Sep 04, 2014	Dec 15, 2015	Sep 04, 2014			
es1	es11	50	Sep 04, 2014	Sep 05, 2014	Sep 04, 2014			
IDEA	JABBA_MONITORING	0.01	Jun 19, 2014	Jun 01, 2015	Jun 18, 2014			

## 7. Developing Wasabi

### 7.1 Using Wasabi's Code style

Wasabi provides us with a code style environment for both IntelliJ and Eclipse. They are obtained in the bin directory and easily imported to the user's IDE. When writing a brand-new code those code snippets should be used to align the addition with the rest of the codebase.

### 7.2 Extending Server-side modules

In Wasabi, the Guice is extensively used for dependency injection. In order to scale the functionalities, one needs to implement the appropriate interfaces, and increase the corresponding module class to bind objects to the implementation, which is optional.

## 7.3 Authentication

- Interface: `com.intuit.idea.abntest.authentication.Authentication`

By implementing this interface, you can include your own authentication scheme. Be sure to update the `<authentication.class.name>` property in the root pom.xml file with your appropriate class.

The default implementation for the Authentication interface is `com.intuit.idea.abntest.authentication.impl.DefaultAuthentication`. It reads user credentials from the `userDirectory.properties` file.

## 7.4 Authorization

- Interface: `com.intuit.idea.abntest.authorization.Authorization`

Implement this interface to include your own authorization scheme. Be sure to update the `<authorization.class.name>` property in the root pom.xml file with your appropriate class.

If you extended the `com.intuit.idea.abntest.authentication.AuthenticationModule` Guice module with your own implementation, then you must extend the `com.intuit.idea.abntest.authorization.AuthorizationModule` Guice module as well in order to install your appropriate authentication module.

By implementing this interface, you can include your own authorization scheme. Be sure to update the `<authorization.class.name>` property in the root pom.xml file with your appropriate class.

## 7.5 Real-Time Data Exportation

The data exportation can be done in real time in wasabi as the events occur. There can be two types of datatypes that can be exported in Wasabi; Assignment data and Impression or action data.

### 7.5.1 Assignment data

As soon as the assignment data is created or modified, it can be exported by creating a class that deals with the Assignment Executor interface and execute when a payload object has been received.

### 7.5.2 Impression/Action data

It can be implemented by modifying the Events' module class methods to install the assignments method classes.

## **8 Case Studies**

### **8.1 Case study 1:**

#### **Removing Promo code boxes from checkout page Increased Total revenue by 24.7%**

**Test completed by:** Bionic Gloves, an online store that outlines and offers a scope of games, military, and easy-going gloves.

**Tested Elements(s):** 'Gift voucher' and 'Exceptional voucher' code boxes on the checkout page.

**Hypothesis:** Including promo-code boxes and Gift vouchers on the checkout page will result in increase of sales and lower cart abandonment rate.

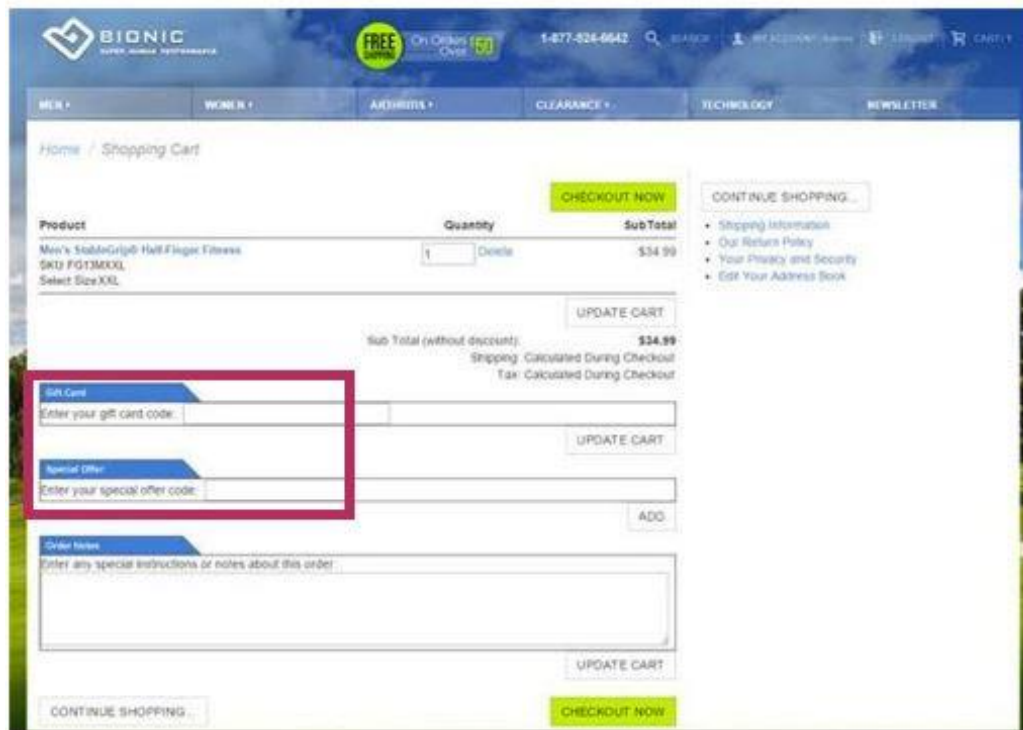
**Test:** The test was carried out on 1400 visitors for a time period of 48 days.

#### **Control Page**

The page before changes are made to the website and the following page is visible to the users in the control bucket.

#### **Experimental Page:**

The page after changes are made on the check-out page and the following page is visible to the users in the Experiment bucket. In this page, metrics like click through rates, and actions performed by the user are navigated and data is collected about each user and it is stored for further analysis.



**Result:** The total revenue increased by a handsome 24.7% and Revenue Per Visitor (RPV) by 17.1%.

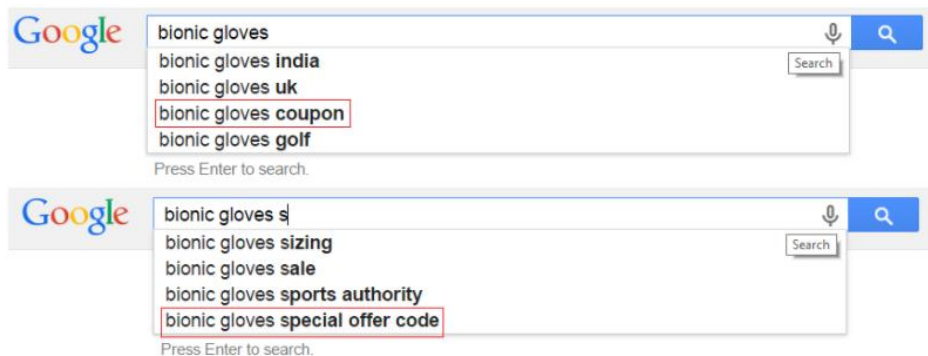
### Analysis:

The promo-code boxes on the original page were leaking customers out of the order process!

On finding the 'Gift voucher' and 'Unique Offer' code boxes just before directing to the payment page, guests were provoked to leave the checkout procedure and look for the rebate loaded promo-codes somewhere else.

### Supporting conclusion to above idea:

The above idea was further supported by Google's auto-complete search for 'Bionic Gloves':



So, people were in fact searching for the promo-codes! The variation offered no such diversion to the visitors and outperformed the control.

## 8.2 Case study 2:

### Displaying Competitors' Higher Prices for Specific Products Made Conversions Jump by 10%.

**Test carried out by:** Paper stone, a UK based eCommerce website that deals in office supplies.

**Element(s) tested:** Price information on product page.

**Hypothesis:** Displaying competitors' higher prices on product pages will increase clicks on 'Add to Basket' and overall website conversion rate.

**Test:** The test was run on more than 12000 visitors for a period of two weeks.

### Control Page

0845 567 4000 (Mon-Fri 9am-6pm) | Contact Us | Dispatch and Delivery | Help and FAQs | Login / Register

Supplying office products to the UK for over 9 years

What does your office need today? Search

My Basket 12 items £415.30 Checkout

Office Supplies | Paper Products | Technology & Machines | Printer Ink & Toner | Furniture | Kitchen, Cleaning & Facilities | Login / Register

Free Next Day Delivery (over £50) | Hassle-Free Returns | Low Price Promise | Award-Winning Supplier | Free Gifts

Back to previous product | Home > Ink & Toner > Epson > Inkjet Printer

**Epson 18XL Inkjet Cartridges High Capacity 31.3ml Black/Cyan/Magenta/Yellow Ref C13T18164010 [Pack 4]**  
Product code: 100189

Original Epson inkjet cartridge  
Total capacity: 31.3ml  
One each of Black, Cyan, Magenta and Yellow  
Packed 4.

Epson's four-colour ink set delivers crisp, clear text documents and glossy, lab-quality photos.

Product specification | See compatible printers

RRP £56.25  
**£39.99** ex VAT  
Save £16.26 (29%)

1 Add to Basket

Next working day delivery (32 in stock)

Add Favourite

Paperstone made two varieties of the control and both were conveying a contenders price module.

Note that this module was given on only 5000 item pages from a sum of more than 18000. Paperstone had the value advantage just on these 5000 items.)

## Variation 1 and Variation 2

This happened presumably in light of the fact that the contender's price module didn't have a differentiating presence on the page. The module was a plain content form and may have turned into a casualty of 'standard visual deficiency'.

**Epson 18XL Inkjet Cartridges High Capacity 31.3ml Black/Cyan/Magenta/Yellow Ref C13T18164010 [Pack 4]**  
Product code: 100189

- Original Epson inkjet cartridge
- Total capacity: 31.3ml
- One each of Black, Cyan, Magenta and Yellow
- Packed 4.

Epson's four-colour ink set delivers crisp, clear text documents and glossy, lab-quality photos.

Product specification See compatible printers

RRP: £56.25  
**£39.99** ex VAT  
Save £16.26 (29%)

Viking	£46.49
Staples	£40.83
Compared on 26/01/14	

1 Add to Basket

Next working day delivery (32 in stock)

Variation 1

**Epson 18XL Inkjet Cartridges High Capacity 31.3ml Black/Cyan/Magenta/Yellow Ref C13T18164010 [Pack 4]**  
Product code: 100189

- Original Epson inkjet cartridge
- Total capacity: 31.3ml
- One each of Black, Cyan, Magenta and Yellow
- Packed 4.

Epson's four-colour ink set delivers crisp, clear text documents and glossy, lab-quality photos.

Product specification See compatible printers

RRP: £56.25  
**£39.99** ex VAT  
Save £16.26 (29%)

1 Add to Basket

Next working day delivery (32 in stock)

**Price Comparison**

Paperstone	£39.99
Viking	£46.49
Staples	£40.83
Compared on 26/01/14 See disclaimer	

Variation 2

The second variation worked much better and prevailed upon the control.

The contender's costs module was energized — Paper stone's cost was featured against the contenders' costs, and a header content was included. The module was likewise repositioned below the 'Add to Cart' button in a bid to clear up the CTA's surrounding space.

With a more reverberating presence now, the module productively showed Paper stone's favourable position and activated transformations.

## **9. Advantages of Wasabi A/B Testing:**

**Useful in low-data rate tests:** If your landing page has only a very less conversions per single day, one essentially can't utilize a further developed tuning strategy.

**Simplicity of implementation:** Most of the software packages support A/B testing. You even might have the capacity to gather the information you require with your current Web analytical tools.

**Simplicity of testing process:** Split tests don't need to be deliberately outlined or adjusted. You just choose what number of versions you need to test and after that split the available traffic evenly among them.

**Ease of results evaluation:** Just few basic statistical tests are needed to decide the winners. By comparing the value of baseline version to each experimental version to see if you've reached your desired statistical confidence level.

**Results are reliable:** The capacity to mix and match enables you to test a scope of developmental and progressive alternatives in a single test, without being compelled by the more granular meaning of factors in a multivariate test.

## **10. Disadvantages of A/B testing:**

**A/B testing measures only quantity, not quality:**

It can just reveal to you how clients are responding to your site, not regardless of whether they are the "right" or best clients to demonstrate your site to.

**A/B testing requires sacrificing some conversions as part of the experiment:**

If numerous version performance is worse than the control, then it leads to accidentally prompted a giving up of various recruits that we would somehow stall out to the control. By Testing various more terrible performing versions and, after some time, the yielded changes would signify a great deal of customer loss.

**Things change:**

Simply because A performed better form version B one year prior, does not imply that it will, in any case, perform better at this point. We ourselves have seen distinctive results when we led a similar test twice amid various circumstances.

Preparing huge no of variations require work. More no of tests results in generation of more content or data possible development, and further maintenance (maintaining a client's website is extremely important, by the way).

The test rounds must be arranged deliberately if an objective is to quantify interaction effects between isolated elements.



## 11. Code Implementation of Wasabi:

```
1 //Configuration
2
3 var server = "abtesting-demo-e2e.a.intuit.com"; //Server Information
4 var appName = "Demo_App"; //Application Name
5 var expLabel = "BuyButton"; //Experiment Label
6 var buckets = [ "BucketA", "BucketB" ]; //Buckets
7 var colors = [ 'DarkSeaGreen', 'Peru' ]; //Colors
8
9 // Experiment related Information
10 var userID;
11 var experimentID;
12 var bucket;
13 //Fetch the user value
14
15 userID = escape($('#userIDext').val());
16 //Ajax Call to the WASABI
17 var assignmentCall = $.ajax({
18     url: "http://" + server
19         + "/api/v1/assignments/applications/"
20         + appName
21         + "/experiments/"
22         + expLabel
23         + "/users/"
24         + userID,
25     dataType: 'json',
26     timeout: 300,
27     success: function(data) {
28         response = JSON.parse(data);
29         bucket = data.assignment;
30     }
31 });
```

The above app displays Green or Blue button based on the user. We created two buckets in our experiment with a sampling rate of 50% each. When the user logs in, an API call to AB Testing is made and fetches the application name, experiment label and user ID of the user in the app.

Whenever the app gets a user id, it makes a call to assignments API of Wasabi. It will evaluate the request and return the bucket which the user belongs to.

After our bucket assignment from WASABI, the application then decides the Experiment to show to the user based on what experience to show based on the bucket.

### Logging impressions:

```
32  
33 curl -v -H 'Content-type: application/json'  
34 -d '{"events":[{"name":"IMPRESSION"}]}'  
35 'http://abtesting-demo-e2e.a.intuit.com/api/v1/events/applications/$appName/experiments/$experimentName/users/$userId'
```

### Logging Actions:

```
curl -H 'Content-type: application/json'  
-d '{"events":[{"name":$actionName}]}'  
'http://abtesting-demo-e2e.a.intuit.com/api/v1/events/applications/$appName/experiments/$experimentName/users/$userId'
```

For the AB Testing service to determine which bucket is the winner, it needs to know how many users saw the experience and if the users took any actions on that experience. Impressions are like page views and actions are button clicks, hovers, right clicks, checkouts, or whatever metric you want to measure.

Wasabi can be embedded using Maven:

```
<dependency>  
  <groupId>com.intuit.wasabi</groupId>  
  <artifactId>wasabi</artifactId>  
  <version>1.0.20160627213750<build_timestamp></version>  
</dependency>
```

## **12. Suggestions:**

Wasabi has been a great tool to perform A/B Testing. Now-a-days, as there is a lot of competition in the market, Wasabi needs to evolve in order to meet the requirements of the end users. Some of the upgrades that can be done on wasabi to enhance its product quality are

### **12.1 Improve the Integration ability of the application:**

One of the major drawbacks of wasabi Tool is that it is currently integrated only with Linux Operating Systems. As most of the Clients use Windows Operating system, it would be better Wasabi can be implemented in Windows. This would help them increase their customer base.

### **12.2 Scalability:**

Currently wasabi runs different experiments. One of the feedback that was given by the end users is to provide a Preview feature, where one can preview the experiment before running the experiment. This would help the customer to cross check the features, before testing it on the platform.

### **12.3 Interoperability:**

Currently Wasabi supports JSON format of exchange. We can expand the scope of wasabi by supporting other communication formats such as XML etc.,

### **12.4 Moving to Cloud Servers:**

Wasabi is a self-hosting platform. It runs on the client's machines only where the A/B Testing is performed. This makes it difficult to host wasabi as the client need to establish the required infrastructure that meets the requirements of wasabi. To avoid this convolution, Wasabi can be transformed into a cloud service. This will reduce the cost of infrastructure, maintenance and improve the accessibility to wasabi services.

Some of the popular cloud hosting platforms like Amazon web services can be used by wasabi.

### **12.5 Automating the wasabi:**

In the current competitive market companies like Google, Microsoft, Amazon has lots of A/B testing experiments to be performed on daily basis. To accommodate many such experiments, it would be difficult using the current wasabi implementation. By incorporating Automation into the 'Wasabi' we can run many experiments in parallel.

One thing that can be done is by providing a feature of uploading the excel where the excel contains the information of experiments, buckets and other segmentation information. This eliminates the process of manual addition of the experiments and hence can automate it to certain extent.

The future scope can be providing automated A/B test scripts which can be used to run to perform the ab testing by eliminating the manual effort.

### **13. References:**

1. <https://intuit.github.io/wasabi/v1/guide/index.html#extending-server-side-modules>
2. <https://intuitblog.com/technology/engineering/meet-wasabi-an-open-source-ab-testing-platform/>
3. <https://www.linkedin.com/pulse/open-sourcing-wasabi-ab-testing-platform-intuit-lucian-vlad-lita/?articleId=6965091159100307371>
4. <https://medium.com/blueprint-by-intuit/the-architecture-behind-wasabi-an-open-source-a-b-testing-platform-b52430d3fd80>
5. [https://developer.intuit.com/docs/0100\\_quickbooks\\_online/0400\\_tools/0040\\_wasabi](https://developer.intuit.com/docs/0100_quickbooks_online/0400_tools/0040_wasabi)
6. <https://www.forbes.com/sites/danwoods/2017/03/09/why-ab-testers-have-the-best-jobs-in-tech/#5804599f73fe>
7. <https://vwo.com/blog/>
8. <http://jamesbachini.com/wasabi-opensource-ab-testing-platform/>
9. <https://github.com/intuit/wasabi>
10. <https://blog.effectiveexperiments.com/product-review-wasabi-an-open-source-ab-testing-framework/>

11. <https://alternativeto.net/software/wasabi/>

12. <https://sdtimes.com/sd-times-github-project-week-wasabi/>