

ADAPTIVE MAIL

TEAM MEMBERS:

<i>S.Suganthan</i>	<i>20203271401232</i>
<i>S.Suthahar</i>	<i>20203271401233</i>
<i>R.B.Tamilarasan</i>	<i>20203271401234(TeamLeader)</i>
<i>S.Tamoas sanjay</i>	<i>20203271401235</i>
<i>G.Vignesh</i>	<i>20203271401236</i>

Under the guidance of

*Mr.V.S.SubramonianM.C.A,M.Phil.,
Assistant Professor*

+

Table of content

Title

1 INTRODUCTION

1.1 Overview

1.2 Purpose

2 PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy Map

2.2 Ideation & Brainstorming Map

3 RESULT

4 ADVANTAGES & DISADVANTAGES

5 APPLICATIONS

6 CONCLUSION

7 FUTURE SCOPE

8 APPENDIX

A. Source Code

1.INTRODUCTION

One of the common actions is to send a confirmation email to the submitter on successful submission of Adaptive Form. To accomplish this we will select the “Send Email” as submit action.

You can use email template or just type in the body of the email as shown in this screenshot below.

Notice the syntax to insert form field values in the email. We also have the option to include form attachments in the email, by selecting the checkbox “include attachments” in the configuration properties.

When the Adaptive Form is submitted, the recipient will get email.

1.1 Overview

Oracle is completing the process of replacing the CRM search technology with Adaptive Search, a high-performance search engine that provides keyword searching and enhanced filtering capabilities. Digital Sales UIs and features rely

exclusively on Adaptive Search. In CX Sales, some work areas still rely on legacy CRM Search technology.

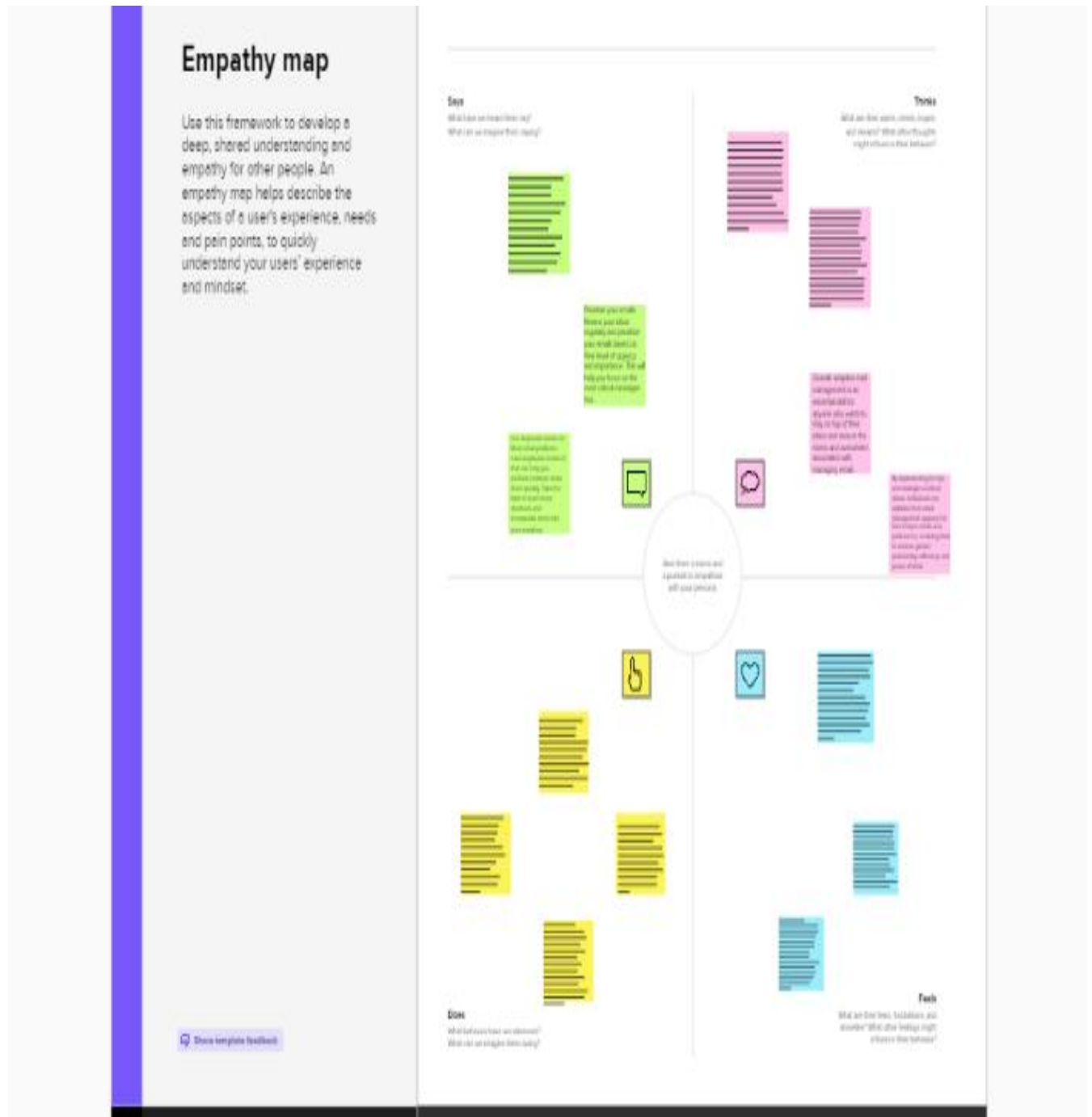
By default, all customers now have Adaptive Search enabled automatically for Accounts and Contacts. You can enable other objects using Setup Assistant or manually. Not all the available objects are sales objects. For example, Partner and Partner Contact are only pertinent for Partner Relationship Management. HR Help Desk Request, Installed Base Asset, Internal Service Request, and Service Request are specific to Fusion Service. Adaptive Search is also available on custom objects or child objects you create.

1.2Purpose

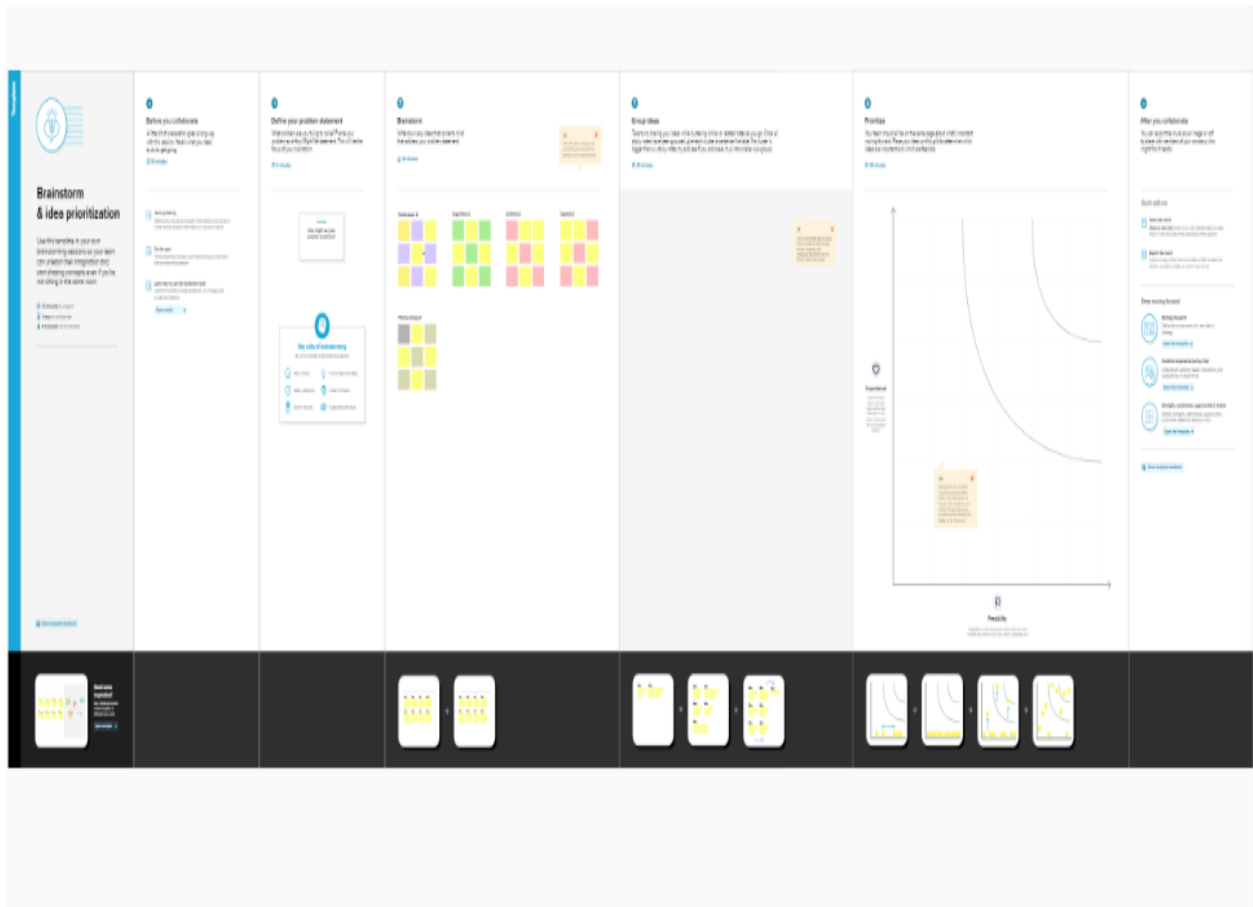
1. The system is "triggered" when the user accesses (and retrieves) a page.
2. A *user model* is used to determine how the page needs to be adapted.
3. The user model is updated, based on the page access.
4. The adapted page is sent to the browser.

2.PROBLEM DEFINITION & DESIGN THINKING

Empathy map



Ideation & Brainstorming Map



3.RESULT



Login

Username

Password

Login

Sign up

Forget password?



Register

Register

Have an account? [Log in](#)

Home Screen



Send Email

View Emails

4.ADVANTAGES & DISADVANTAGES

Advantages

- Cost- effective - the costs of email marketing can be much lower than many other forms of marketing. There are no advertising fees, printing or media space costs.
- Permission-based - your marketing list will be made up of people who have actively chosen to receive email communications from you. Customers who are genuinely interested in your products and/or services are more likely to engage with your business.
- Flexible design - you can send plain text, graphics or attach files - whichever suits your message best. A choice of design options gives you scope to convey your business branding.
- Personalisation and segmentation - with email marketing you can personalise messages. You can also segment your marketing list, so that your customers receive messages from you that they are interested in - this will help boost their engagement with you.

Disadvantages

- Spam - commercial email or 'spam' irritates consumers. If your messages aren't targeted to the right people, the recipient may delete your email or

unsubscribe. You need to make sure that your email marketing complies with privacy and data protection rules, and that it is properly targeted at people who want to receive it. The 'click through rate' for untargeted emails is likely to be very low. See [email marketing and privacy law](#).

- Undelivered emails - poorly designed emails may not get delivered. Emails that use certain spam keywords or characters in the subject heading or content of the email, egFREE, click here, are likely to be filtered out by email software and internet service providers. If you don't keep your marketing lists up to date, you will find incorrect email addresses mean your messages won't reach the right person.
- Design problems - your email must be designed so that it appears as it should across multiple devices and email providers. You may encounter a trade-off between design and functionality. Some people opt to receive text-only emails, consider how your message will look if this is the case.

5. APPLICATION

Adaptive Mail app is a sample project that demonstrates how to use the Android Compose UI toolkit to build a conversational UI. The app simulates a messaging interface, allowing the user to send and receive messages, and view a history of previous messages. It showcases some of the key features of the Compose UI toolkit, data management, and user interactions.

- Users register into the application.
- After registration , user logins into the application.
- User enters into the main page
- User can View previously sent emails.
- User can give subject and email body to send email.

6. CONCLUSION

In this paper we described a technique automatic filtering of messages, and the architecture of a system that implements it. Our technique, based on clustering algorithms for data classification, was tested in several experiments, showing a high degree of flexibility, efficiency and effectiveness in the message classification context.

The current implementation has to be considered only as a starting point of the general framework of message filtering. Indeed, several issues have still to be addressed.

- The first issue is concerned with the classification of incoming messages.

Currently, the tool works fine a large amount of mail messages: that is, we assume a model of interaction in which the user periodically re-organizes all his e-mail messages.

This type of interaction does not take into account the problem of updating the model, as described by the points 2 and 3 in page 4. Such problems have to be addressed, by combining the proposed approach with supervised learning techniques.

7. FUTURE SCOPE

When creating websites that work well on a range of screen sizes and devices, responsive web design is far and away the most popular approach.

[Recommended by Google](#) and embraced by many web designers and developers, it is, however, not the only solution for creating multi-device friendly websites.

While not as popular as responsive web design, an adaptive approach is another viable solution for developing multi-device-friendly sites.

In reality, creating websites that accommodate a variety of possible screen sizes is not only about [responsive versus adaptive](#). The future of multi-device web design is rather a combination of both.

8.APPENDIX

Source Code

```
<?xml version="2.0" encoding="UTF-8"?>
<project version="4">
  <component name="AutoImportSettings">
    <option name="autoReloadType" value="NONE" />
  </component>
  <component name="ChangeListManager">
    <list default="true" id="419f213c-6c58-4852-803e-ea540ddf9e8a"
name="Changes" comment="" />
    <option name="SHOW_DIALOG" value="false" />
    <option name="HIGHLIGHT_CONFLICTS" value="true" />
    <option name="HIGHLIGHT_NON_ACTIVE_CHANGELIST" value="false" />
    <option name="LAST_RESOLUTION" value="IGNORE" />
  </component>
  <component name="CodeInsightWorkspaceSettings">
    <option name="optimizeImportsOnTheFly" value="true" />
  </component>
  <component name="ExternalProjectsManager">
    <system id="GRADLE">
      <state>
        <projects_view>
          <tree_state>
            <expand />
            <select />
          </tree_state>
        </projects_view>
      </state>
    </system>
  </component>
  <component name="FileTemplateManagerImpl">
    <option name="RECENT_TEMPLATES">
      <list>
        <option value="Kotlin Data Class" />
        <option value="Kotlin Interface" />
        <option value="Kotlin Class" />
        <option value="XML Properties File" />
      </list>
    </option>
  </component>
  <component name="KotlinCodeInsightWorkspaceSettings">
    <option name="optimizeImportsOnTheFly" value="true" />
  </component>
  <component name="MarkdownSettingsMigration">
    <option name="stateVersion" value="1" />
  </component>
  <component name="ProjectId" id="20E08r4iH28fygZinGCd0lwF3vF" />
  <component name="ProjectViewState">
    <option name="hideEmptyMiddlePackages" value="true" />
    <option name="showLibraryContents" value="true" />
  </component>
  <component name="PropertiesComponent"><![CDATA[{
```

```
"keyToString": {  
    "RunOnceActivity.OpenProjectViewOnStart": "true",  
    "RunOnceActivity.ShowReadmeOnStart": "true",  
    "RunOnceActivity.cidr.known.project.marker": "true",  
    "cidr.known.project.marker": "true",  
    "settings.editor.selected.configurable": "editor.preferences.tabs"  
}  
}]></component>  
  
<component name="RunManager">  
    <configuration name="E:/adaptive mail app" type="GradleRunConfiguration"  
factoryName="Gradle" temporary="true">  
        <ExternalSystemSettings>  
            <option name="executionName" />  
            <option name="externalProjectPath" value="$PROJECT_DIR$" />  
            <option name="externalSystemIdString" value="GRADLE" />  
            <option name="scriptParameters" />  
            <option name="taskDescriptions">  
                <list />  
            </option>  
            <option name="taskNames">  
                <list />  
            </option>  
            <option name="vmOptions" />  
        </ExternalSystemSettings>  
  
<ExternalSystemDebugServerProcess>true</ExternalSystemDebugServerProcess>  
  
<ExternalSystemReattachDebugProcess>true</ExternalSystemReattachDebugProcess>  
        <DebugAllEnabled>>false</DebugAllEnabled>  
        <method v="2" />  
    </configuration>  
    <recent_temporary>  
        <list>  
            <item itemvalue="Gradle.E:/adaptive mail app" />  
        </list>  
    </recent_temporary>  
</component>  
  
<component name="SpellCheckerSettings" RuntimeDictionaries="0" Folders="0"  
CustomDictionaries="0" DefaultDictionary="application-level"  
UseSingleDictionary="true" transferred="true" />  
    <component name="TaskManager">  
        <task active="true" id="Default" summary="Default task">  
            <changelist id="419f213c-6c58-4852-803e-ea540ddf9e8a" name="Changes"  
comment="" />  
            <created>1681110282690</created>  
            <option name="number" value="Default" />  
            <option name="presentableId" value="Default" />  
            <updated>1681110282690</updated>  
        </task>  
        <servers />  
    </component>  
  
<component name="XDebuggerManager">  
    <breakpoint-manager>  
        <breakpoints>  
            <line-breakpoint enabled="true" type="kotlin-field">  
                <url>file://$PROJECT_DIR$/user.kt</url>  
                <line>10</line>
```



```
        <properties myFieldName="firstName" myClassName="User" />
        <option name="timeStamp" value="1" />
    </line-breakpoint>
</breakpoints>
</breakpoint-manager>
</component>
</project>
```