Comment report

Lists all the questions in the survey and displays all the comments made to these questions, if applicable.

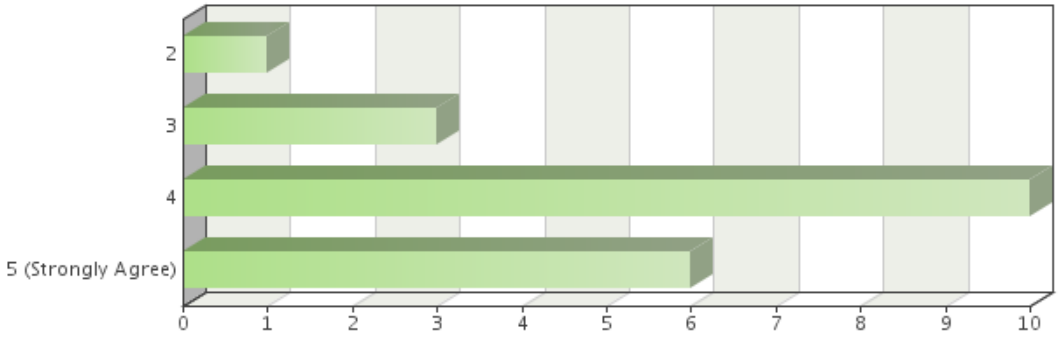
Table of contents

Report info

Report date:	Friday, April 24, 2015 4:53:52 PM EST
Start date:	Friday, September 5, 2014 1:56:00 PM EST
Stop date:	Thursday, December 31, 2015 1:56:00 PM EST
Stored responses:	20
Number of completed responses:	20

Question 1

It was easy to use RAPPT.



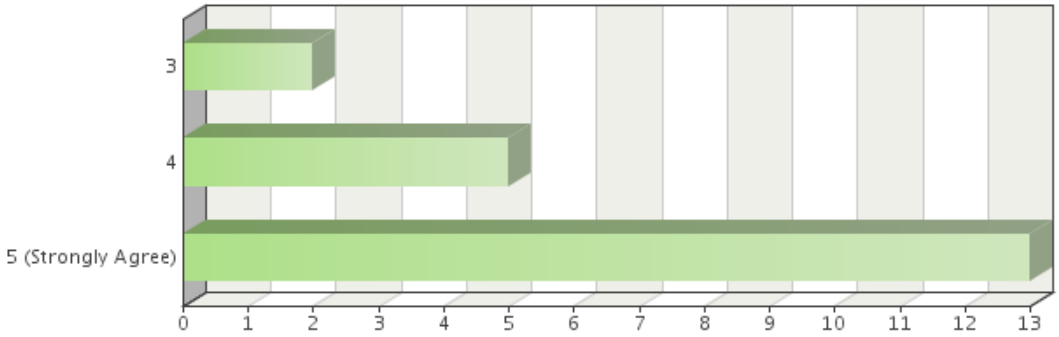
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
2	1	5%	5%
3	3	15%	15%
4	10	50%	50%
5 (Strongly Agree)	6	30%	30%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 2

It is easy to understand what each icon represents.



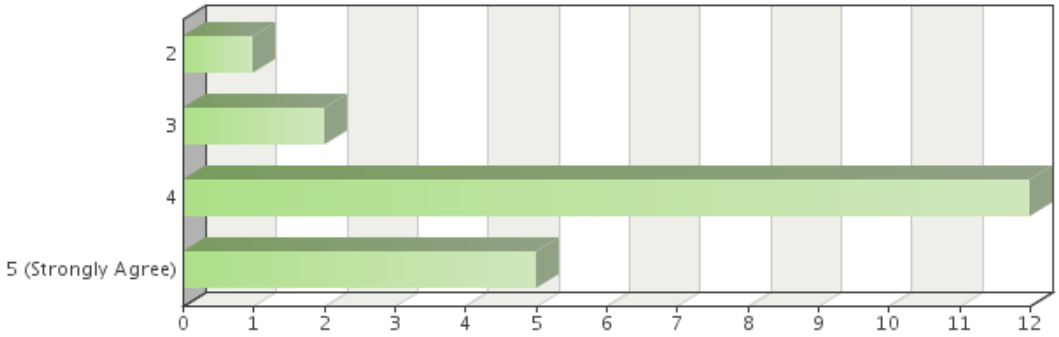
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
3	2	10%	10%
4	5	25%	25%
5 (Strongly Agree)	13	65%	65%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 3

It was easy to learn how to use RAPPT.



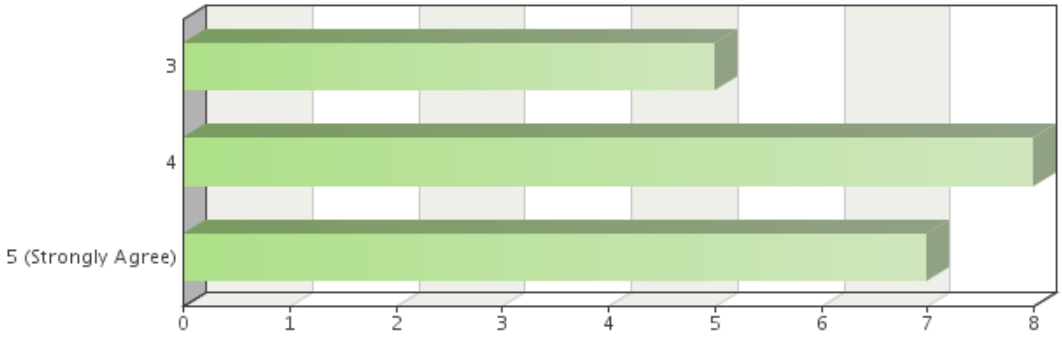
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
2	1	5%	5%
3	2	10%	10%
4	12	60%	60%
5 (Strongly Agree)	5	25%	25%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 4

It was easy to create a screen with a list.



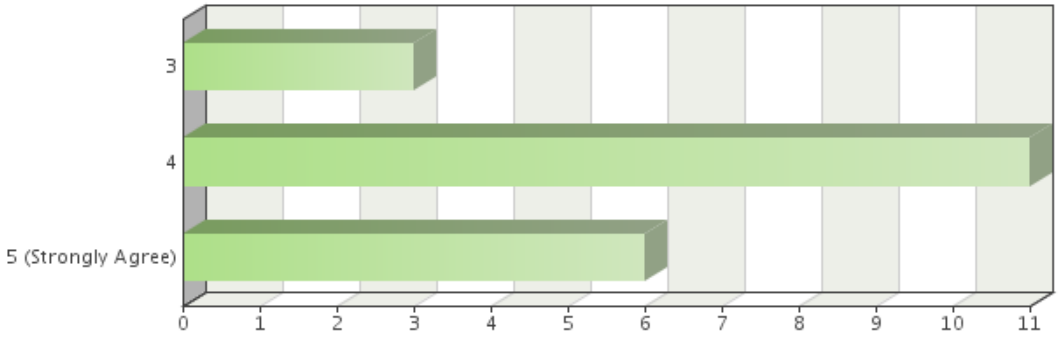
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
3	5	25%	25%
4	8	40%	40%
5 (Strongly Agree)	7	35%	35%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 5

It was easy to load data from an API and render it to the screen.



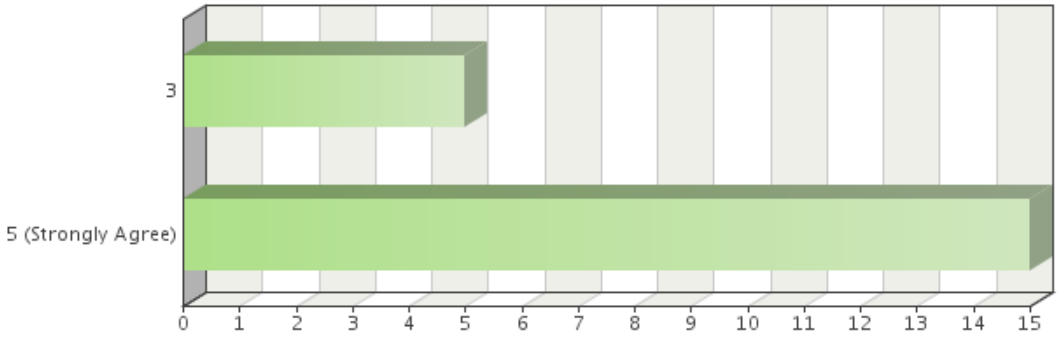
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
3	3	15%	15%
4	11	55%	55%
5 (Strongly Agree)	6	30%	30%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 6

It was easy to add a tabbar to the app.



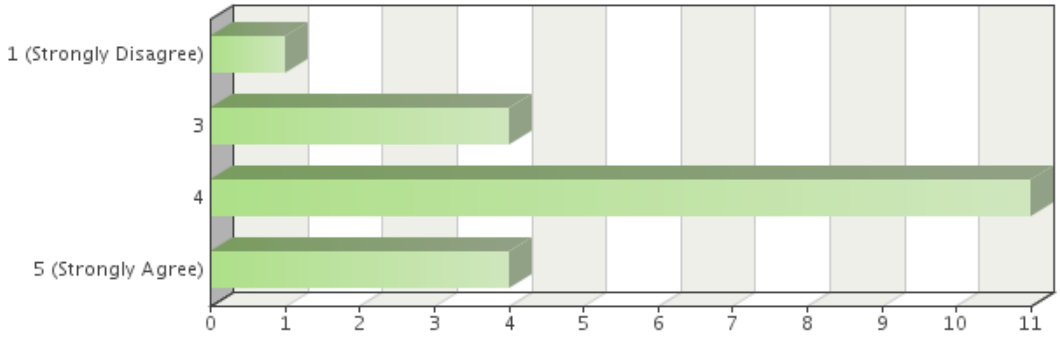
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
3	5	25%	25%
5 (Strongly Agree)	15	75%	75%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 7

It is easy to remember how to use RAPPT.



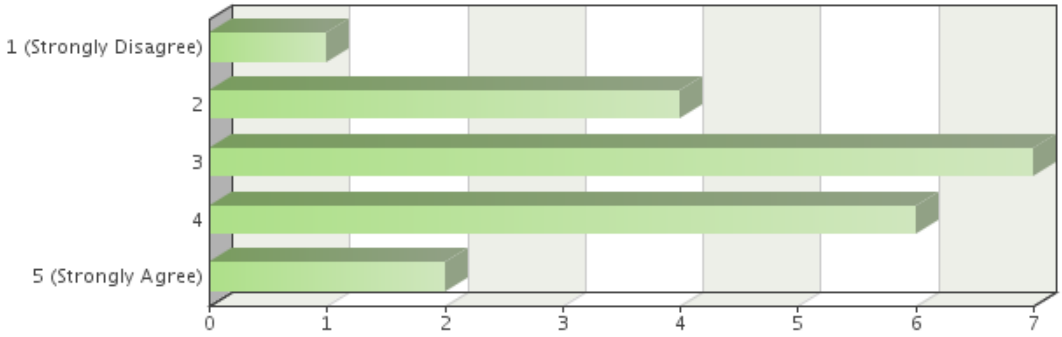
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
1 (Strongly Disagree)	1	5%	5%
3	4	20%	20%
4	11	55%	55%
5 (Strongly Agree)	4	20%	20%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 8

It is easy to avoid making errors or mistakes.



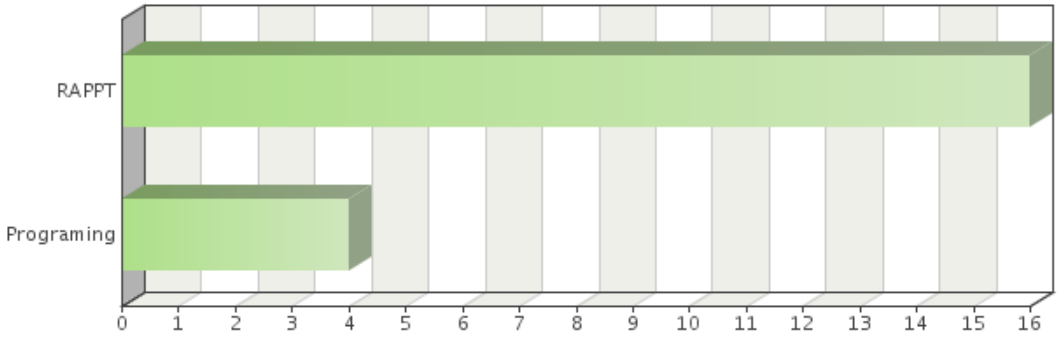
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
1 (Strongly Disagree)	1	5%	5%
2	4	20%	20%
3	7	35%	35%
4	6	30%	30%
5 (Strongly Agree)	2	10%	10%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 9

If you had to start to develop an app again what would you prefer to use?



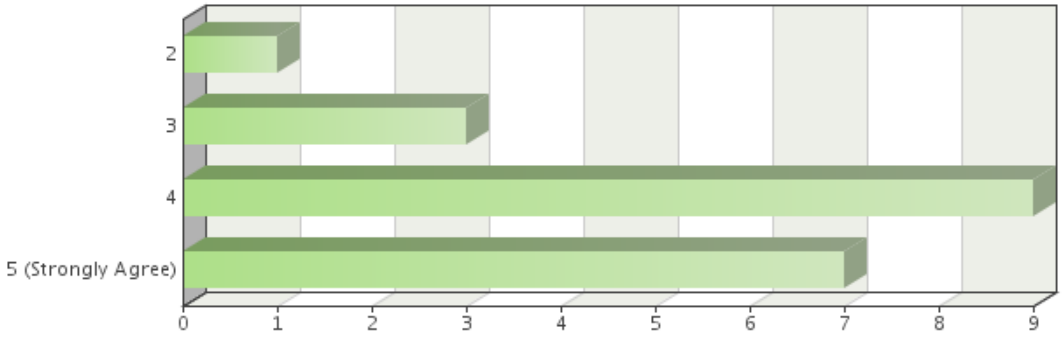
Frequency table

Choices	Absolute frequency	Relative frequency	Adjusted relative frequency
RAPPT	16	80%	80%
Programing	4	20%	20%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 10

RAPPT makes you more productive when starting a new project than using templates from an IDE.



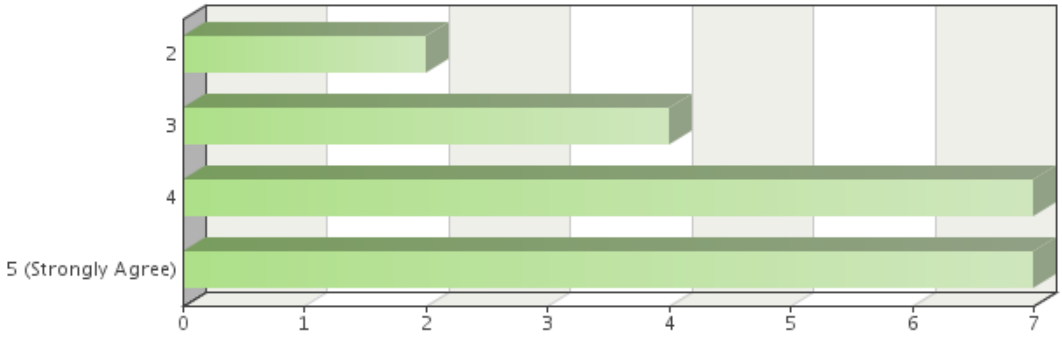
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
2	1	5%	5%
3	3	15%	15%
4	9	45%	45%
5 (Strongly Agree)	7	35%	35%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 11

RAPPT helps you avoid common errors when starting a new project.



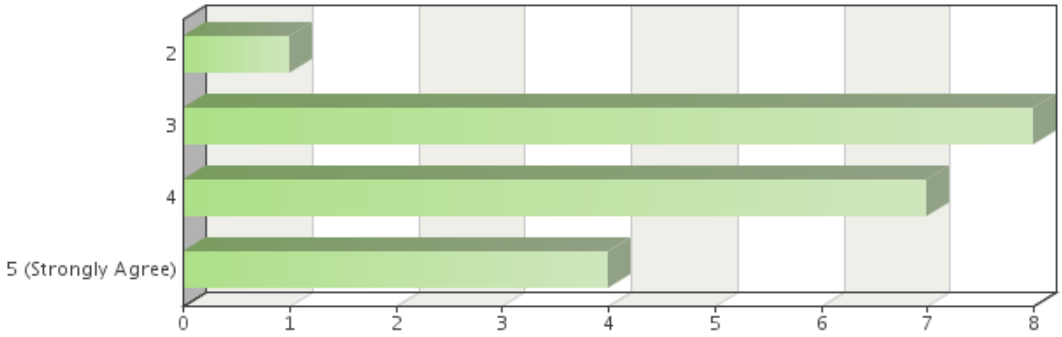
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
2	2	10%	10%
3	4	20%	20%
4	7	35%	35%
5 (Strongly Agree)	7	35%	35%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 12

The concepts in RAPPT are sufficient for modelling a mobile app.



Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
2	1	5%	5%
3	8	40%	40%
4	7	35%	35%
5 (Strongly Agree)	4	20%	20%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 13

What concepts would you like to see included in the visual model?

Text input
more fine grained toolbox and page development UIs, for example would like to have the labels in the drag and drop page.
Live preview of how it will be looked when app is installed
Theming.
More objects (like buttons and what not), that you can drag into the screens
Drawer Toolbar Buttons Browser
A way to setup the layout of the individual screen.
Designer for screens to change the look and feel of the apps.
some further usage of its drag-n-drop to perhaps have visual data models, or even some rudimentary element positioning within the screens
More visual components
I would like to see more UI components like button, checkbox etc.
A list placeholder just for design purposes
Video and Audio concepts.. drag and drop audio/video files
APIs Controls Datasource
Android Emulator so as to be able to visually see and interact with the output
More difficult to add UI components such as scrolling full page text, rich text content, side swipe menus etc

Question 14

What concepts would you like to see included in AML?

Text input
perhaps I can specify APIs, data sources more abstractly now that the code gets generated automatically.
syntax auto complete
Module composition and plugins.
Maybe realtime checking for simple errors (like for method brackets not closed), which underlines the tet
I expected curly brace pairs to create lexical scopes. Data sources in separate screens should be able to have the same name. Optional naming for things I don't care about (e.g. view groups, list rows, tab bars, & labels with no data binding).
Toolbar Time picker Dialogs
more input types, although there could very well be support for this already and I am just unaware. the ability to have one screen and switchable data sources
syntax for adding more components
I would like to see code suggestions
Maybe intents? eg. grab a photo or something might be useful for apps with a "social" component.
background app servies
So far so good
Live syntax highlighting / errors. Syntax suggestions Compile -> visualization in the cloud

Question 15

What would you like to see in the visual model that is already in AML?

Text input
label specification and other UI elements. perhaps bindings as elements to drag and drop?
list declaration and initiation (row content), so then in AML we can just connect data model with it
Rendering of view elements.
Something to signify activities are linked via a mutual tabbar, so separate areas of the app with different tabbars can be differentiated (with colours for eg)
For each screen, a representation of the views it contains. Some sort of representation of the API, and how each screen interacts with its endpoints.
Drawer Buttons
A way to setup which variables are being passed between screens.
Some more ability to create connections. Perhaps even a visual data model that relations can be created between.
Editing text in the visual model for individual screens
I would like to have the ability to make connections between UI elements and other screens. For example, connection between tab bar and other screen.
Lists, as noted above Also apparently drawers are a thing
APIs Controls
Visual indicators for the tab bar navigation that can be turned on and off
Get the code only for a screen or other component when you double click. Binding of events.

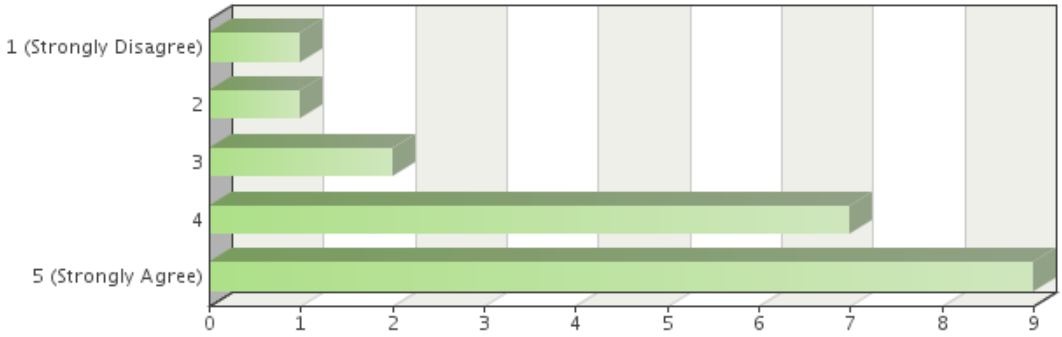
Question 16

What would you like to see in AML that is already in the visual model?

Text input
I would prefer the visual elements over AML! Move everything to visual elements please!
From what I see AML contains everything in the visual model?
Nothing really!
I don't think there really is anything in the visual model that I think should be in AML, the visual model felt very "drag this screen in, now go to the AML and actually do things".
Not sure
Nothing specific because AML is more advanced than visual model
Is this a trick question?
So far so good
-

Question 17

Using RAPPT is more efficient than starting with a raw Android project.



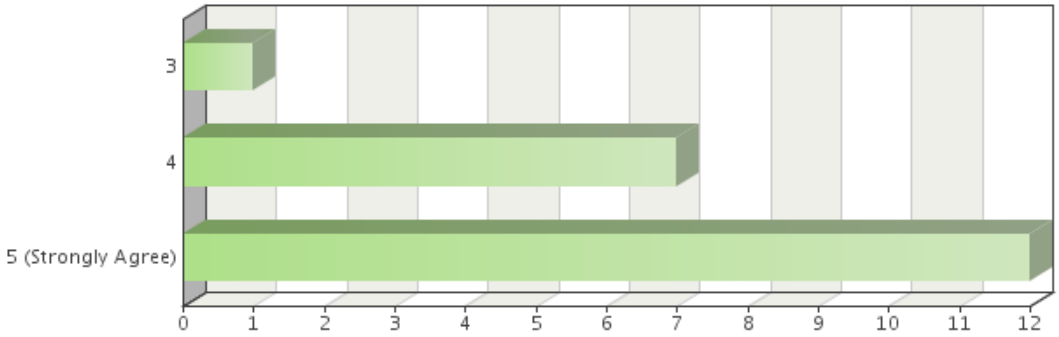
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
1 (Strongly Disagree)	1	5%	5%
2	1	5%	5%
3	2	10%	10%
4	7	35%	35%
5 (Strongly Agree)	9	45%	45%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 18

RAPPT is useful for mobile app development.



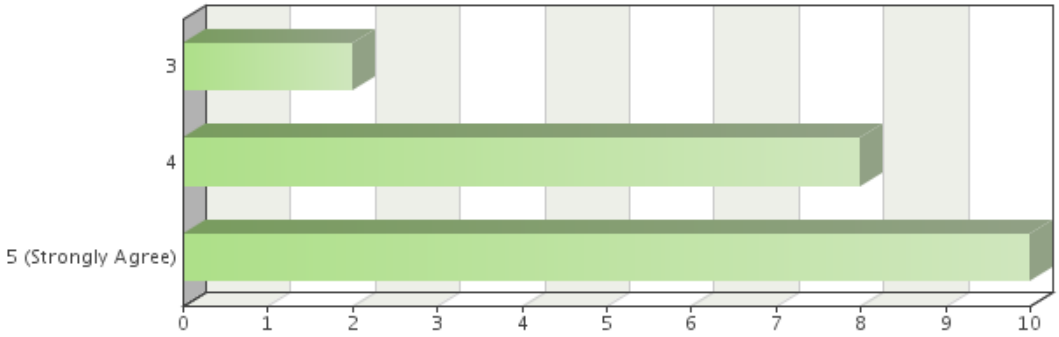
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
3	1	5%	5%
4	7	35%	35%
5 (Strongly Agree)	12	60%	60%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 19

You are satisfied with using RAPPT.



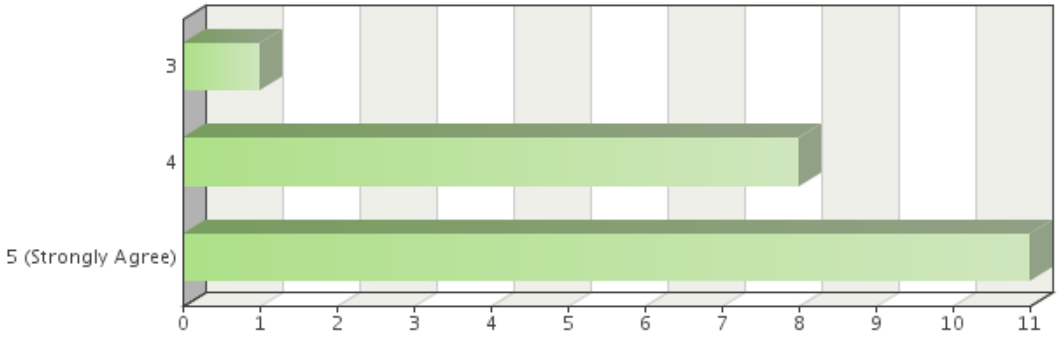
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
3	2	10%	10%
4	8	40%	40%
5 (Strongly Agree)	10	50%	50%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 20

You would recommend RAPPT to a friend.



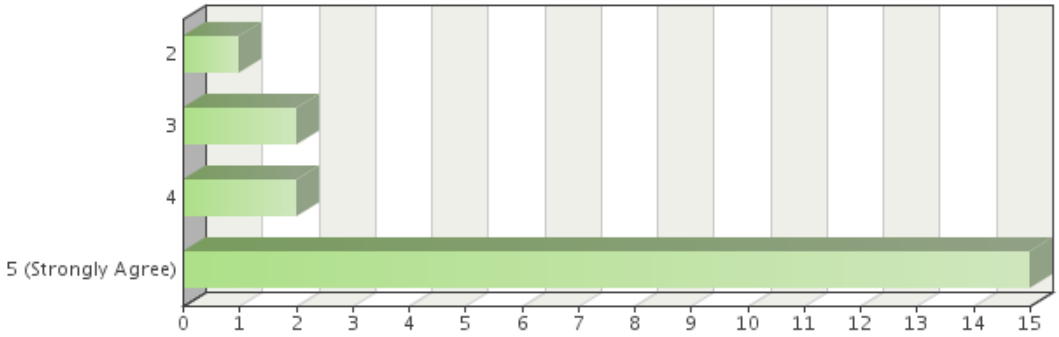
Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
3	1	5%	5%
4	8	40%	40%
5 (Strongly Agree)	11	55%	55%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 21

It was fun to use.



Frequency table

Levels	Absolute frequency	Relative frequency	Adjusted relative frequency
2	1	5%	5%
3	2	10%	10%
4	2	10%	10%
5 (Strongly Agree)	15	75%	75%
Sum:	20	100%	100%
Not answered:	0	0%	-

Total answered: 20

Question 22

In your opinion, which types of apps are not suited to be developed with RAPPT?

Text input
cannot think of any!
Stand alone app (no need to connect to any server)
Highly visually customised apps.
Apps with a single or two screens that require lots of information and data manipulation
Games with complex interaction
Those with complex UIs, many background services, and that interact with non-RESTful APIs.
Games Browser Chromecast based Media based
Larger apps with more complicated functionality requirements.
Games, and apps which require customised UI elements or are very complex
Custom apps, that don't follow standard Android design. As far as I know, I don't know if I could include, for example, maps or custom elements using RAAPT.
Anything that is not data driven probably wouldn't be ideal, for example a game isn't feasible. Also if there are any cases where you want to use libraries (for example Fabric.io) to display data there is obviously no support.
Not sure
I think games are not suited to be developed with RAPPT
Non-data driven apps eg. games, or apps that integrate more strongly with the Android OS (camera, intents)
Gaming and cmpplex
Games Offline/online Database
Fully scoped projects.
Complex apps.
Applications that require lots of user interactions, such as phone game.

Question 23

What did you like the most about RAPPT?

Text input
the drag and drop of pages to start development, the lack of previous knowledge of API and app development
It's able to produce ready to use app instantly
The high-level abstraction and terseness of the code.
I was able to create, compile and run a project with no trouble, with a headache
The website is pretty and the DSL straight-forward.
Speed of getting an app set up
The simplicity of setting up the core elements of an app.
Quick and has examples
AML, over using the the IDE. It was much faster to create new screens and tabs using AML.
I liked the combination between the visual model and the AML to create the app
the ease and simplicity saves a LOT of time (seriously!)
The interface is easy to use and the AML language is easy to learn
The model block was a really nice way of specifying API bindings, with the arrows clearly showing the flow of data.
Ease of creating project and integratin Apis
Easy to use, save time!
It minimised boiler plate code significantly.
Lowers the barrier of entry if you dont have a programming mindset
Models and views are separated.

Question 24

What did you dislike the most about RAPPT?

Text input
The error reposting could be improved. Perhaps the code generation and installing could be included as part of the tool as well, not a separate Andriod studio app
No immediate feedback via the web (besides compilation) that what you're building is working.
The designer screen didnt have many objects
Inconsistent AML concepts
The random auto formatting. My OCD does not like code being out of line and all over the place.
Wasn't obvious to what extent I could use APIs (without looking through multiple examples), maybe provide an examples pane that only shows examples on the item you're working on
I kept trying to save my code... which doesn't work in-browser.
I felt that the visual model was really only around to quickly create the AML for the screens. There was no way to remove the tabbar from a screen, it had to be changed in the AML. AML felt it missing reference of possible structs.
visual model sometimes appears buggy
Nothing specific
There isn't really a way to see if the app actually works without having to tangle with Android Studio; I'm not sure if that's a sticking point for actual Android devs but if a Rappt -> AS -> Rappt cycle develops during dev that might be annoying.
Compile error validations could be more helpful
Many things could be done from the Visual Editor and make life more easier
Without a means to view the output, it can be tricky to determine if runtime errors will be encountered. It helps to at least test for configuration of the API endpoints to text the link between results and labels.
Seems a bit backwards, i.e tabs added to the pages (assuming I did it correctly) not the main controller. Really needs code completion / hints to assist you when typing in AML.
Sometimes the compiling messages are not clear enough to identify the errors in codes.

Question 25

What needs to be added or fixed for you to use RAPPT for all your data intensive Android app projects?

Text input
more fine-grained elements in the tool box to be used in page element development, less need for writing code (I am biased here!)
App browser can be resize (cannot see items when it is too wide)
Modularity and the idea of evolving a codebase, as opposed to just using it as a once-off bootstrapper.
More visual objects for the designer screen!
The samples need descriptions. It was difficult to see (at a glance) how they differed. It is unclear as to how arguments passed to screens are bound. For example, "alpha2Code" (in the samples) is not explained. More IDE features. Auto-completion being the primary one.
- Tab spacing is inconsistent (generated is 4 spaces, tab is 2 spaces) - Response field name conflicts with keywords (results.title -> screen.title) - Conflict in endpoint naming and screen naming (even with _ placed in front of screen name) - Naming conflict with datasource in different screens. You shouldn't need to have datasource0, datasource1, etc for different screens
Improvements to the titles of the example AML code. They were not entirely clear without clicking in each of them what they were for.
Saving / Retrieving (doesn't have to be cloud based) Predictive code completion / hints would be nice too.
Custom element design via RAPPT.
I don't feel like anything needs to be fixed, however I do believe that if some of the features/concerns stated in previous questions were addressed it might make the experience a little crisper.
More components to the visual model would be a good addition Not able to suggest any fixes at the moment as I need more time to play around with Rappt to determine any fixes
The ability to use the local database.
* Changing id of landing page in Designer doesn't update AML (presumably model isn't updated) * Variable scoping is kind of unclear, esp. in regard to the navigate-to statement: it's not obvious that you don't bind a variable via the model block to pass to the next screen (for me anyway)
Running Background services, Camera integration, Audio integration,
Bidirectional update (AML/Visual), Apps reverse engineering
^ Output emulator as mentioned multiple times, although it is clearly out of scope for what this platform is intended to achieve.
Compile + Debug so you can see it work online / or locally if integrated.
Provide more details about "internal error".