

FOR INVENTORS

The non-technical skills for building apps





Lenovo Scholar Network

MOBILE APP DEVELOPMENT PROGRAM



Introduction	1
The Four Meanings of "Design"	2
Empathy	3
Ideation	4
Designing Your App	6
Storyboards	7
Wireframes	8
Paper Prototypes	9
Scoping Your Project	11
Team Collaboration	12
Developing Your App	14
Development Cycle	14
Last Known Good State	15
Intellectual Property	16
Copyright and Creative Commons	17
Presenting Your App	18

Created in partnership by NAF and MIT App Inventor. This work is licensed under <u>https://creativecommons.org/licenses/by-sa/3.0/</u> Building apps is not only about coding. Successful app development teams work through all of the following phases:

- 1. Empathy
- 2. Ideation
- 3. Designing Your App
- 4. Scoping Your Project
- 5. Developing Your App
- 6. Presenting Your App

Coding takes place during the development phase, and a significant amount of thought, planning, and effort goes into an app before, during, and after the coding happens. This document is a guide to many of the non-technical steps of building an app. These steps are important and can make or break the development of an app!

Although it is tempting to jump directly into coding, it is important to take the time to define and design your solution with clear documentation and communication between team members. Good planning will lead to a more efficient work process and, ultimately, **a better product**.

Much of the success of your app will also be determined by the ability of your team to work well together. Some questions to consider: Does your team distribute work appropriately? Do you communicate effectively? Does everyone have the same goal in mind? This guide also includes some tips for organizing your team and work during the development phase.

After reading this document, you should be well prepared to embark on your app building project and be ready for any setbacks that may occur during the development of your project.

Good luck, fellow inventors!







Before getting into the steps involved in building an app, let's discuss what we mean by 'design.' When we speak of 'design,' there are four meanings we could be referring to:

- 1. Graphic design the images, colors, and visual look
- 2. The User Interface (UI) how the appearance and graphic design choices support use of the app
- 3. The User Experience (UX) how a person uses the app
- 4. The overall plan the concept of the app

So, when someone tells you to 'design an app,' what does that actually mean? For the Lenovo Scholar Network contest, it means all of the above.





noun

the action of understanding, being aware of, being sensitive to, and vicariously experiencing the feelings, thoughts, and experience of another of either the past or present without having the feelings, thoughts, and experience fully communicated in an objectively explicit manner.

The first step to creating an app is identifying a problem to solve or a need to meet. To empathize with someone is to put yourself in their shoes, to understand their problem fully. Empathizing with someone else's problem helps you to understand their need and how your app can fill that need. If you fully understand the user's need, it will give you better insight into possible ways to address the issues in your app. Learn as much as you can about your user. Talk to them, interview them, observe them in their daily lives. For example, imagine that you're thinking about an app to reunite dog owners with lost dogs. Find a dog owner, or three. Ask them if they've ever lost a dog, and if so, how did they feel? What steps did they take to find their pet? What worked? What didn't? What were their frustrations? The next stage of designing the app is **ideation**, the formation of ideas and concepts about how your team will address the problems or opportunities identified during the **empathy stage**. Brainstorming is a great way to generate many ideas. Let your mind go, come up with as many ideas as you can, no matter how unconventional they may be.



Here's a way to organize your **Ideation** work:

Materials needed:

- sticky notes
- pens
- timer

Designate someone as the leader of this activity. They can take care of the timer, tell everyone when to start and stop each step, and lead your discussions.

1. Start by better **understanding the problem you are trying to solve.**

Discuss what insights and information you gathered during the **Empathy** stage. Clearly break down the problem your users are trying to solve by answering these questions:

- a. Who are your target users for this app?
- b. What are their goals and motivations?
- c. What are their primary concerns or barriers to achieving their goals?
- 2. The next step is idea generation. Give each team member a stack of sticky notes and a pen. Set a timer for five minutes, during which time all team members should write down as many ideas as they can come up with on the sticky notes (one idea per note). Then post them up on a large board or wall.

The more ideas, the better. The wall should be covered with sticky notes. Don't hold back! Any idea you come up with, you post. Make sure everyone is writing notes. Encourage team members to be as inventive as they can. This is the time for creativity and thinking outside the box, so quantity outweighs quality for this exercise. There is no communication between team members during this activity. This is a time for each person to contribute their own ideas.

3. Next is **idea grouping**. Set the timer for five minutes again. This is still a silent activity. All team members should go up to the board of ideas and move sticky notes around to group them by similar topics. For instance, if you see two game ideas, move them next to each other. Two lawnmower trackers? Move them next to each other!

This is also the opportunity for everyone to read each other's ideas. This might also generate new ideas. If so, write them down and group them appropriately. Note which ideas you like, which one's appeal to you.

4. Finally, move on to idea refinement. No more being silent! Talk to your fellow team members. The activity leader can direct people's attention to the larger groups of notes. This can start the discussion of what ideas multiple people had, and which ideas gel for the team. Are there large groups of notes that relate? Could they morph into a more refined app idea? Are there ideas that you particularly like? The activity leader should get everyone involved in the discussion, eliminate ideas that could be too outside the box, unappealing to the group, or too big to reasonably be developed in the time allowed. Then start to focus on the ideas that resonate with the group, and that people are excited to talk about.

DISTILLING IDEAS INTO A SOLUTION

The end product of this brainstorming process should be a short document that identifies:

- Your audience
- The problem or opportunity your app will address
- How the app will address the problem or opportunity

5

After identifying your problem and deciding on what your solution will be, it is time to make the solution more concrete by starting to plan out the app. You want to visualize what the app will look like, the user interface or UI, as well as how it feels to the user as they interact with the app, the user experience or UX.

"

Designing [is a] conversation with the materials.

-Donald Schön, MIT Professor in urban planning

"

While it's tempting to jump right into MIT App Inventor and begin dragging buttons and labels in the Designer, it's better to start with paper. Using paper and pencil for these steps has several advantages. It's quick, gets the idea across quickly, and can help to clarify or even generate more ideas. You can easily push out more iterations of your paper prototypes after the team, and more importantly after your end user, can look at them and give you feedback. Use the templates at the back of this document for your own storyboarding, wireframing, and paper prototyping.

STORYBOARDS

This technique is often used when people make movies to visually describe the action taking place, scene by scene. For your app, you can use storyboards to tell the story and flow of user interaction in your app. It should focus on user tasks: What makes them use the app in the first place? What steps do they take when using the app? What are the intended results?

FIND MY DOG VIDEO Storyboard template Planning the filming of the promotional film





Susie goes out to get her dag & sees the dog Ler the End MY DOG gpp is gone! Starts to weep. ち





due

with Ler

look for her dog



FIND M DOG ne LOR

he.

helps

form

Mavia

out the

7

WIREFRAMES

Wireframes are a first step in the design layout of your app. It provides the general structure of the different components of the app, without providing too much detail or design elements like logos or images. The purpose is to show how all the elements fit together on the screen, and how navigation or flow will happen in the app. At this stage simplicity is key!



PAPER PROTOTYPES

The paper prototype is the next step. You visually show what the user interface will look like on the screen. With the prototype, you can see actual UI elements - buttons, labels, images, etc. Your app is starting to take shape, and you can see more details. You could show this to a user or set of users to get feedback. The prototype itself can be like a storyboard, where the steps a user takes can be visualized and the flow understood.





9-

These design steps help to clarify your ideas within the team and to show end users what the app will look like. It is a good idea to have potential users of the app try out the user interface at the paper prototype stage, before you even build the app. Cut out each drawn screen so it's a separate piece of paper. You can then ask your users to walk through a scenario using the app, treating the piece of paper as if it were the real app. As the designer, you can respond to their interactions by placing the appropriate screen in front of them based on what they do. For example, if your user presses on the "Lost Dog" button by touching it on the paper, replace that piece of paper with the "Lost Dog" screen and let them interact with that. Let the user direct what happens and see if they can figure out what to do. This process allows for feedback on the feel and flow of the user interface, and can alert you to any problems with the design or confusion about how the elements work in the app. And since this happens on paper, you will avoid wasting precious development time if things need to be changed. When embarking on a large project like making a mobile app, an important consideration is how ambitious your solution is. Maybe the idea is a good one but will take more time than is available. Maybe the idea simply isn't possible with available technology, no matter how much time you spend on it. Be realistic about your ideas, and about your ability to make your ideas into a working app.

A good practice when starting to plan out your app is to list all the features you want your app to have. Then identify which ones are **core features**—meaning the app will not work or be complete without them. For example, with the app to find lost dogs, a beautiful user interface would be great. However, it is not as important as adding a map that shows where to find lost dogs. Identify your **minimum viable product**, which is a version of your app containing the most important features to early users. Once you have this version you can show users the concept and see if it satisfies their needs. You can use their feedback to iterate and continue to develop your app.

Take your list of features and prioritize them by how important they are to your core goals for the app. You should also give them weights according to how long it might take to incorporate those features in the app. Balancing important features against the amount of time it will take to include those features will help to define and order the features you add to your app.

FEATURE	PRIORITY	EST. TIME TO COMPLETE
Login screen	3	
Main menu (2 buttons)	1	
Lost dog entry screen	1	
Found dog entry screen	1	
Lost dogs display	2	
CloudDB (internet storing)	3	
Automatic notification to owner	2	

FEATURES PRIORITY LIST

Team collaboration can make or break a project. Every team member should have a defined role (or roles) and should understand what that role is (designer, developer, researcher, marketer, etc). It may be important to have a team leader as well. For this project, the team leader could be called the **project manager**.



DOES:

- assign tasks to other members
- check on progress of tasks
- contribute to final product in other roles
- communicate with other team members

DOES NOT:

- act bossy
- hover over other team members
- play favorites

The job of the project manager is not bossing the rest of the team around. The project manager assigns tasks and checks on the progress that the other members are making on their tasks. The project manager is also a member of the team and almost always contributes to the final product in other roles. The project manager's role is to keep the big picture in mind, be aware of all the components of the app, and to track the progress of all the components.

The team, led by the project manager, should develop a project timeline, so that everyone's time is optimized and there are not long periods of waiting and not working. The project manager should constantly communicate with team members to make sure everyone is on track and that the app development is not affected by unforeseen circumstances or a missed deadline. For example, if José is designing the user interface and can't get the user feedback to confirm the design, that means that Marisa, who is implementing the design in App Inventor, cannot start her part of the project. There are many software tools that exist to help manage a complex project like building an app. One example is <u>trello.com</u>, where each team member can share their progress with each other, and you can track which tasks are dependent on others.

If one member cannot accomplish their task until another member accomplishes theirs, then you have a **single point of failure**, where the project can stall until that part of the project can be completed or fixed.

DEVELOPMENT CYCLE

The period when the team is working on designing and building your app is called the development cycle. **Agile software development** is an approach to software development in which a team works towards a set of goals for a project, adjusting and modifying based on changing needs and situations.

The **Scrum Framework** is a model for doing agile development. The idea is to set a list of priorities, then set the task to complete that list in a designated amount of time, called a **Sprint**, while meeting regularly (usually daily) to check in on everyone's status in meeting goals. During the meeting, the Scrum Master, who could be the teacher or the project manager for the team, should ask 3 questions of each team member:

- What did you accomplish yesterday (or since we last met)?
- What do you hope to accomplish today (or before we next meet)?
- What is blocking your progress?

Once one Sprint is complete, the cycle starts all over again. It is good practice if each Sprint results in a product that works and can be shown to users for feedback. This is called the **Minimum Viable Product**, or MVP. The Scrum framework keeps everyone on the team focused on a set of goals, and in constant communication.

SPRINT GOALS	TO DO	IN PROGRESS	DONE
Found Dog Screen			Jenna: Copy Lost Dog Screen components
		Jenna: Complete user interface in Designer	
	Martin: Coding blocks		
	Gina: Testing with emulator		
	Ariana: testing with mobile phone, tablet		
	Ariana: Meet with clients to get feedback		

LAST KNOWN GOOD STATE

Development should happen in stages. You can't build a whole app in one day. Most importantly, you should always try to have a **Minimum Viable Product** (MVP). That way, if you are asked for an impromptu demo or have to stop development without warning, you will have something to share. Also, if something breaks or goes wrong, you have a **last known good state** to return to—a lot like your computer does when it runs into an unrecoverable error.

Checkpoints:

When you're at a MVP stage in App Inventor, where you've added a feature or features that are working as you want, make a checkpoint. This saves your project in its current state as a separate project, but then lets you continue to edit or update your current project.

Projects 👻	Connect 🗸	Build 🗸	Help 🗸
My projects			
Start new project Import project (.aia) from my computer Import project (.aia) from a repository Delete Project			
Save project Save project as Checkpoint			
Export selected project (.aia) to my computer Export all projects			computer
Import keystore Export keystore Delete keystore			

This way, you can have multiple copies of your project, at each new feature stage.

Checkpoint - LostDogApp	
Checkpoint name:	LostDogApp_MVP1
Cancel	ОК

INTELLECTUAL PROPERTY

Most likely, you will want to use images and other media in your app. This is where **copyright** comes in.

There are three primary categories of copyright:

- Only available for use with purchase (i.e., radio stations can only play songs they have purchased the rights to, or stock photos available for purchase from Shutterstock). You must pay to use these.
- Available for use with attribution (i.e., many photos available on Flickr—the artists often allow you to use their photographs as long as you credit the source).
- 3. Freely available for any use and modification. (The **modification** bit is important you need to make sure you're free to change it to fit your needs without restriction. Or, at least, you need to understand what the restrictions are.)

Just because you found a media file, does not mean you have the permission to include it. For example, if you want music in your app, you can't just add any song, even if you bought the song for your personal use. This would be a copyright violation. Similarly, if you find an image on Google Images, it does not mean you have permission to use it in your app. It may surprise you, but most memes are also protected by copyright. Someone created the underlying image, and they own the copyright. Just because something has gone viral, does not mean it's legal to use it.

The easiest solution is to create your own content. You own the copyright for anything you create. Need a drawing of a panda? Draw it yourself! Need a cat meow sound effect? Record your own cat! (Be sure to get permission from your cat first, or anyone else you record.)

COPYRIGHT AND CREATIVE COMMONS

However, it's not always feasible to create your own content. This is where **Creative Commons** (CC) comes in. Anything with a CC license can be used without having to worry about copyright. You can even find some memes on Creative Commons. Most CC media are of category 2 or 3, from above.

There are two ways to find Creative Commons media:

- 1. Use the official CC search: <u>https://search.creativecommons.org/</u>
- 2. When using Google Images or Bing Images, you can filter by license.



Sample Images Search from Google.com

Other good sources for images and icons that have large volumes of royaltyfree options are Pixelbay, IconFinder, and FontAwesome. You've built your app - congratulations! Now, it's time to show the world what you've done. You need to **engage** your audience. You need to come up with a compelling story to tell so that your audience cares about what you've done, and wants to see more or maybe use the app themselves. What original problem are you trying to solve? What is your solution? Who does your app help? How does it work? What makes your solution work better than other competitors? Telling the story behind your app can help. For example, perhaps you chose to make a dog finding app because you lost your family pet and it was a traumatic experience. That story can be the hook that generates interest in your app. You can find examples from previous Lenovo Scholar Network winners at <u>lenovoscholars.com</u>.

It is helpful to build a **storyboard** to outline what you want to tell your audience. Outline each part, and then fill it in with text and drawn images that you will eventually turn into a slide presentation or video. You can use the same template you used to storyboard your app.

Storyboard template $\mathcal{F}(N\mathcal{I} \cap \mathcal{M}) \cap \mathcal{D} \cap \mathcal{C}$ $V \cap \mathcal{D} \in \mathcal{O}$ Planning the filming of the promotional film



Susie goes out to get her dag & sees the dag is gone! Struts to weep.



call when another user finds how dog



Her friend Maria shows Ler the EndMyDos app to look for her day,









STORYBOARD FOR:



WIREFRAME FOR:





Lenovo Scholar Network MOBILE APP DEVELOPMENT PROGRAM

Follow Us on Social Media

@NAFCareerAcads
@LenovoScholars

@MITAppInventor

<u>naf.org</u> <u>lenovoscholars.com</u>

Created in partnership by NAF and MIT App Inventor. This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License <u>(http://bit.ly/2Yey0k5)</u>