

Lomiri App Development Level 1

Book 1 in a series of 3

Lomiri App Development Level 1

Brave New Books

©© This book is released into the public domain using the CC BY-NC-SA 4.0 code.
To view a copy of the CC BY-NC-SA 4.0 code, visit:

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Colophon

This document was typeset with the help of KOMA-Script and L^AT_EX using the kaobook class.

Publisher

First printed in March 2026 by Brave New Books

ISBN: 9789465389240

When you are trained to be a consumer. Community driven development can be strange at first glance. However the power to stand upon the shoulders of giants, and to contribute back to the community is an necessary act of Goodness to save our Freedom of Choice.

– Ir. Erik Mols

Preface

This book is based upon the Ubuntu Touch app development book 1, written by Koen Vervloesem. The book is adapted to the latest distributions and implemented as a Course by OS-SCI. This actual course is converted to a book, which you are looking at now.

About video content

Some sections in the original course contained embedded videos. In this book, each video is represented by a QR-code, you can scan the QR code to watch the original video.

About all the URL's

This book is full with all kind of references to websites and other resources. All URL's are put at the bottom of each page for your convenience.

About OpenSource Science B.V.

This second version of this book is created by OpenSource Science B.V. (OS-SCI)¹

More Than a Motto, It's Our Mission

Freedom First means empowering everyone to:

- ▶ Learn without barriers
- ▶ Share without restrictions
- ▶ Innovate without proprietary limits

It's not just about software — it's about a global culture of openness and trust that leads to better technology and a better world.

Bridging the Open Source Talent Gap

At OpenSource Science, our mission is to bridge the gap between the world's reliance on open source software and the shortage of skilled open source developers. From smartphones to space exploration, open source powers the innovations that shape our future — yet too few students are taught how to contribute to this ecosystem.

We believe education is the key to change. By equipping the next generation of developers with real-world open source skills, we empower them to collaborate, solve problems, and innovate the way modern tech companies do. Our mission is to ensure that the future of technology is not only built on open source, but also built by a thriving, diverse community of open source contributors.

Tilburg March 2026 OS-SCI

¹ OS-SCI was founded in december 2022, by Jeroen Baten, Jeroen Bots and Erik Mols, to teach about OpenSource technologies, and train the next generation of FOSS engineers. Visit <https://os-sci.com>

Contents

Preface	v
Contents	vii
1 Introduction	1
1.1 Introduction	1
1.1.1 Introduction to Lomiri App Development	1
1.1.2 Why Ubuntu Touch Matters for Open-Source Innovation	1
1.1.3 Course Support Information	2
1.2 An Introduction to Mobile App Development for Ubuntu Touch	2
1.2.1 Learning Results	3
1.2.2 Exam Information	4
2 Install Ubuntu Touch	7
2.1 Introduction	7
2.2 Troubleshooting	8
3 Lesson 1: Getting your development environment up and running	9
3.1 Introduction	9
3.2 Installing clickable	9
3.2.1 Linux	10
3.2.2 Windows	12
3.2.3 macOS	14
3.2.4 Intel Macs	14
3.2.5 Deploying to Ubuntu Touch from VMware	15
3.3 Hello World	15
3.3.1 Using ADB	18
3.3.2 Using SSH	20
3.4 Installing Visual studio	21
3.5 Getting help	23
4 Lesson 2: Starting a QML project with Clickable	25
4.1 Introduction	25
4.2 Starting from the QML Only template	25
4.3 Project Structure	26
4.4 The Main.qml file	27
4.5 Adding a Button	32
4.6 Adding a Text field	33
4.7 Adding a List	33
5 Lesson 3: Designing your Ubuntu Touch app with QML	39
5.1 Introduction	39
5.2 Using the system theme in your app	39
5.3 Exploring QML components	41
5.4 Positioning components in a row	43
5.5 Adding popup dialogs	45
5.6 Adding leading actions to the list items	49

5.7	Adding an action bar	51
6	Lesson 4: Programming basics in QML and JavaScript	55
6.1	Introduction	55
6.2	Data types, variables and properties	55
6.2.1	Basic Types in QML	56
6.3	Attributes of QML Object Types	58
6.3.1	The id Attribute	59
6.3.2	Property Attributes	59
6.3.3	Method Attributes	60
6.3.4	Signal Attributes	60
6.3.5	Signal Handler Attributes	61
6.4	Java script expressions in QML	62
6.5	JavaScript basics	63
6.5.1	String	63
6.5.2	Number	64
6.5.3	Boolean	64
6.5.4	Array	64
6.5.5	Arithmetic operators	65
6.5.6	Assignment operators	65
6.5.7	String operators	65
6.5.8	Comparison operators	66
6.5.9	Logical operators	66
6.5.10	Ternary operator	67
6.6	Removing items from the shopping list	68
7	Lesson 5: Downloading and storing data	77
7.1	Introduction	77
7.2	Using JSON as a data format	77
7.3	Downloading data with an XMLHttpRequest	79
7.4	The basics of an SQL database	82
7.5	Using an SQLite database in QML with LocalStorage	85
8	Lesson 6: Creating a backend for your app in Python	91
8.1	Introduction	91
8.2	Installing Python	91
8.2.1	Linux	92
8.2.2	Windows	92
8.2.3	macOS	92
8.3	Creating a backend for your shopping list app with Python and Flask	93
8.4	Running your backend on a public server	98
8.4.1	Hosted options	98
8.4.2	Self-hosting on a VPS	99
9	Lesson 7: Your app and the UBports community	101
9.1	Introduction	101
9.2	A short history of the UBports project	101
9.3	The advantages of Ubuntu Touch	102
9.4	Taking part in the community	103
9.5	How to behave in the UBports community	105
9.6	Talk about Ubuntu Touch and your app	106
9.7	Getting inspiration for your apps	107

9.8 Learning by reading code	107
10 Lesson 8 Releasing your app on OpenStore	109
10.1 Introduction	109
10.2 Choosing a license for your app	109
10.3 Version control with Git	110
10.3.1 Install and configure Git	111
10.4 GitLab CI/CD	116
10.5 Publishing your app on OpenStore	117
10.6 Maintaining and updating your app	122
10.6.1 Forking Your Repository	123
10.6.2 Cloning the Fork Locally	123
10.6.3 Creating Changes in a Branch	123
10.6.4 Creating a Merge Request	124
10.6.5 Merging a Merge Request	125
10.7 And much more...	127
11 Lesson 9 Code Quality and Standards	129
11.1 Code Quality	129
11.2 Human Interface Guidelines	129
12 Lesson 10 App Research	131
12.1 Introduction	131
12.2 Dekko	131
13 Assignments	133
13.1 Overview	133
13.1.1 Sample App Ideas for Your Assignment	133

1.1 Introduction

1.1.1 Introduction to Lomiri App Development

Ubuntu Touch, which uses Lomiri as an Operating Environment, is a mobile version of the Ubuntu operating system, designed for touchscreen devices such as smartphones and tablets. Developed and maintained by the UBports community, it emphasizes open-source principles, user privacy, and device longevity. A standout feature is “convergence,” which allows your phone to function as a full desktop when connected to a monitor—bridging the gap between mobile convenience and desktop productivity.

1.1.2 Why Ubuntu Touch Matters for Open-Source Innovation

Ubuntu Touch stands as a beacon for those who believe in the power of open-source technology to democratize software development and empower users. This innovative platform builds on the shoulders of giants, leveraging decades of open-source advancements to create a mobile experience that is both powerful and principled. At its core, Ubuntu Touch uses the Lomiri desktop environment, a project developed and coordinated by the UBports community, which ensures a seamless and intuitive user interface optimized for touchscreen devices. Beyond Lomiri, the UBports community is also responsible for Halium, a groundbreaking project that enables Linux-based operating systems to run on mobile hardware. Halium’s influence extends far beyond Ubuntu Touch—it is the foundation for nearly all Linux mobile operating systems today, demonstrating the community’s commitment to collaboration and shared progress. Unlike traditional mobile ecosystems, Ubuntu Touch fosters a collaborative environment where developers, educators, and enthusiasts come together to shape the future of mobile computing. By choosing to develop for Ubuntu Touch, you’re not just building apps, you’re contributing to a movement that prioritizes transparency, user control, and sustainability. This course is more than a technical guide; it’s an invitation to join a global community dedicated to redefining what mobile technology can achieve. Whether you’re a student, a professional, or simply curious about open-source development, Ubuntu Touch offers a platform where your creativity can thrive, and your contributions can make a lasting impact. Let’s build the future of mobile computing together.

1.1	Introduction	1
1.1.1	Introduction to Lomiri App Development	1
1.1.2	Why Ubuntu Touch Matters for Open-Source Innovation	1
1.1.3	Course Support Information	2
1.2	An Introduction to Mobile App Development for Ubuntu Touch	2
1.2.1	Learning Results	3
1.2.2	Exam Information	4

At OS-SCi, we collaborate closely with the UBports community to offer this comprehensive course as the foundational guide for Ubuntu Touch app development. Structured into 10 practical lessons. Each lesson builds essential skills, from setting up your environment to publishing your app:

- ▶ Lesson 1: Getting your development environment up and running
- ▶ Lesson 2: Starting a QML project with Clickable
- ▶ Lesson 3: Designing your Ubuntu Touch app with QML
- ▶ Lesson 4: Programming basics in QML and JavaScript
- ▶ Lesson 5: Downloading and storing data
- ▶ Lesson 6: Creating a backend for your app in Python
- ▶ Lesson 7: Your app and the UBports community
- ▶ Lesson 8: Releasing your app on OpenStore
- ▶ Lesson 9: Code Quality and Standards
- ▶ Lesson 10: App Research

Lomiri Ubuntu Touch apps are primarily built using QML for the user interface and JavaScript, Python or C++ for logic, with Clickable as the main tool for compiling, testing, and publishing. This course is designed to be accessible to beginners while providing depth for more experienced developers. By the end, you'll have the knowledge and confidence to create your own apps for a platform that values freedom, privacy, and innovation.

1.1.3 Course Support Information

This course is designed as a practical, hands-on tutorial. Although all code and examples have been thoroughly tested, you may occasionally encounter issues—particularly with tools like the Android ADP, which can sometimes be finicky and may not recognize every phone model.

Throughout the course, you'll find clear explanations and step-by-step instructions provided in both written and video formats. We also include live coding demonstrations in the videos to guide you through the process.

If you run into any questions or challenges, don't hesitate to reach out to the community for support. If you're participating in a guided version of this course, your peers and professor are also great resources for assistance.

1.2 An Introduction to Mobile App Development for Ubuntu Touch

Clickable is the main tool to build apps on Ubuntu Touch. It's an easy-to-use command-line tool with a lot of great features. Clickable allows you to compile, build, test, and publish your app, all with one tool. It also provides various templates to get you started with app development.

This course teaches you how to create your own Ubuntu Touch apps with Clickable. You'll develop a shopping list app using QML (Qt Modeling Language) to design its user interface and JavaScript to define its behaviour. Another technology, which can be used for developing Ubuntu Touch apps is SNAP. In the context of this course, we are not discussing SNAP apps. SNAP apps are discussed in our Lomiri development course.

In the rest of this course, we suppose you have a working Ubuntu Touch device. If not, have a look at [the list of supported Ubuntu Touch devices](https://devices.ubuntu-touch.io/)^I. You can install Ubuntu Touch on supported Android devices with the [UBports installer](https://devices.ubuntu-touch.io/installer/)^{II} or buy a [Volla Phone](https://volla.online/) or [Volla Phone X](https://volla.online/)^{III} with Ubuntu Touch pre-installed.

This second version of the course/book, is built upon the first edition.

1.2.1 Learning Results

By the end of this course, you will be able to:

1. Set up and configure a development environment for Ubuntu Touch app development.
2. Create and manage a QML project using Clickable, the primary tool for building Ubuntu Touch apps.
3. Design user interfaces for Ubuntu Touch apps using QML, following best practices for mobile and convergent experiences.
4. Apply programming fundamentals in QML and JavaScript to implement app logic and interactivity.
5. Download, process, and store data efficiently within your app, ensuring data persistence and reliability.
6. Develop a backend for your app using Python, enabling server-side functionality and data management.
7. Engage with the UBports community, including contributing to open-source projects, seeking feedback, and collaborating with other developers.
8. Package and publish your app on the OpenStore, following the submission guidelines and requirements.
9. Apply code quality standards and best practices to ensure maintainable, efficient, and secure code.
10. Conduct app research, including analyzing user needs, exploring existing solutions, and planning features for future development.

^I <https://devices.ubuntu-touch.io/>

^{II} <https://devices.ubuntu-touch.io/installer/>

^{III} <https://volla.online/>

1.2.2 Exam Information

Practical Exam: Lomiri App Development Duration: 2 weeks

Objective: Design, develop, and document a functional Ubuntu Touch app that demonstrates mastery of key concepts from the course. This exam covers 60% of the learning goals (Goals 1–6).

Exam Assignment: Build a "Task Tracker" App

Task: Develop a Task Tracker app for Ubuntu Touch that allows users to:

- ▶ Create, edit, and delete tasks (e.g., to-do items with titles, descriptions, and due dates).
- ▶ Store tasks locally (using QML's LocalStorage or a simple Python backend).
- ▶ Display tasks in a list with a clean, user-friendly interface.
- ▶ Include at least one additional feature of your choice (e.g., task categories, priority levels, or a progress bar).

Requirements: Your submission must include:

1. Functional App:

- ▶ Built with QML/JavaScript (frontend) and optionally Python (backend).
- ▶ Uses Clickable for building and testing.
- ▶ Follows Ubuntu Touch design guidelines (convergence-ready UI).

2. Documentation:

- ▶ A README file explaining:
 - How to install and run the app.
 - Key design choices (e.g., why you structured the QML components a certain way).
 - Challenges faced and how you solved them.
- ▶ Code comments explaining complex logic.

3. Short Video Demo (max 3 minutes):

- ▶ Show the app running on an emulator or device.
- ▶ Demonstrate all core features (adding, editing, deleting tasks).

Criteria	Weight	Details
Functionality	30%	App works as described, with no critical bugs. Additional feature is implemented creatively.
Code Quality	20%	Clean, well-structured QML/JavaScript/Python. Follows best practices (e.g., modular components).
UI/UX Design	15%	Intuitive, visually appealing, and convergent-friendly (scales to desktop).
Data Management	15%	Tasks are stored and retrieved correctly (local storage or Python backend).
Documentation	15%	README is clear, concise, and technical. Code comments explain non-trivial logic.
Community Engagement (Bonus)	5%	Shared progress in the UBports forum or GitLab (e.g., asked questions, contributed to discussions).
Video Demo	15%	Clearly demonstrates all features; professional and concise.

Table 1.1: Evaluation Criteria

Assessment Criteria (100 points total)

Learning Goals Covered The following learning goals are assessed in this exam:

- ▶ Goal 1: Development environment setup (Clickable).
- ▶ Goal 2: QML project structure.
- ▶ Goal 3: UI design with QML.
- ▶ Goal 4: QML/JavaScript logic.
- ▶ Goal 5: Data storage (local or Python backend).
- ▶ Goal 6: Backend integration (optional but encouraged).

Not Covered (for future projects):

- ▶ OpenStore publishing (Goal 8).
- ▶ Advanced community collaboration (Goal 7).
- ▶ App research (Goal 10).

Submission Instructions

- ▶ Code: Push to a public GitLab/GitHub repo (include README.md).
- ▶ Video: Upload to a platform (e.g., YouTube, PeerTube) and share the link.
- ▶ Deadline: [Insert date].

Why This Exam? This assignment tests practical skills while allowing creativity. It mirrors real-world app development, from ideation to documentation, and prepares students for contributing to open-source projects.