e-Channel

Moodle Unplugged

Empowering Hard-to-Reach Learners in Ontario

PREPARED FOR CONTACT NORTH | CONTACT NORD
BY COLLABORATIVE MEDIA GROUP
WITH DR. ISHAN ABEYWARDENA

December 2021

Ontario

Note: The views expressed in this report are the views of Contact North | Contact Nord and do not necessarily reflect those of the province. The Government of Ontario is in no way bound by any recommendations contained in this document.

CONTENTS

PREAMBLE

Contact North | Contact Nord 's (CN) e-Channel Offline Pilot has been developed to address and investigate the needs of learners in communities that lack adequate and consistent Internet access. CN's comparative review of Chromebooks, Rumie, Internet hubs/sticks and Moodle Box technologies provides an overview of recently tested technology solutions.

PROJECT SCOPE

Offer a solution which meets both end-user (learner) needs and the needs of course providers who partner with CN.

DELIVERABLES

Provide written descriptions which would enable the solution to be deployed, namely:

1. A written document summarizing the needed technology processes and supports required to deliver and support a solution to the end user.
2. A written document summarizing the work needed to get a course from a provider's system and ready for use. This document should explain:
    a. how these materials can be updated and maintained;
    b. how assessments (both formative and summative) can be managed when the end-user is not connected to broadband; and
    c. how user identity management is to be managed.
3. A written document outlining the step-by-step processes by which the end-user device could be "wiped clean" and made ready for the next user.

TIMELINES

The work undertaken proceeded in these stages:

1. An initial summary was provided outlining:
    a. which systems or solutions were being reviewed;
    b. an outline of the basic solution being proposed; and
    c. a summary of some of the challenges which the solution needs to overcome.
2. A completed report and analysis, including all the documents outlined in section 3 above was prepared.

PROJECT METHODOLOGY – DESIGN THINKING

Given the unique requirements and challenges of the project, a design thinking approach (Figure 1) was used to formulate the problem statement, identify the stakeholders, define the project scope, understand the limitations of the solution, design the solution architecture, create the prototype, usability testing and real-world implementation.

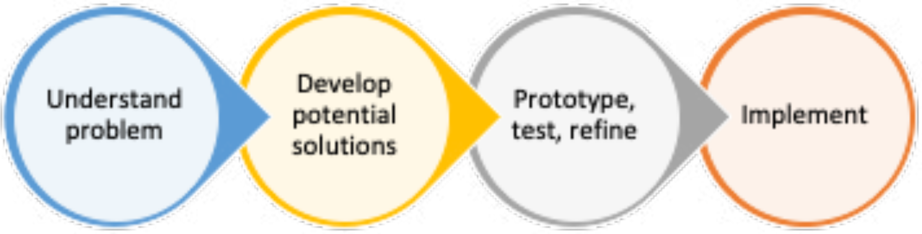

Figure 1 Design Thinking[1]

Understanding the Problem

The process of understanding the problem was threefold: (a) identifying the background by studying the requirements of RFP CN-CN RFP 2021-01; (b) consultations with key stakeholders; and (c) study of the current online courses, technologies and platforms used by the target learners.

Stakeholder Consultations

1. Several initial consultations were held to confirm the requirements and expectations of CN viz. the RFP CN-CN RFP 2021-01, followed by email clarifications on specific points.
2. Secondary consultations were held to confirm the geographical, demographical, and technological composition of the target learners.
3. A consultation with an online course instructor was organized by CN where the consultant was provided with additional course specific details of the requirements, such as the use of H5P.
4. The consultant was provided access to four online courses which represent the courses taken by the target stakeholders. The courses were:
    a. Take a Test (no H5P)
    b. Getting a Jump on Healthy Living (no H5P)
    c. Gearing up for Apprenticeship (H5P)
    d. Numeracy at Work: Ratios & Proportions (H5P and on blended site)

[1] Adapted from https://mitsloan.mit.edu/ideas-made-to-matter/design-thinking-explained (September 2021)

Problem Statement

As a result of the threefold process of understanding the problem, the following problem statement was formulated.

Ideal situation

- All students in Ontario have equitable access to hardware, software and bandwidth which will facilitate uninterrupted end-to-end online learning utilizing various educational technology platforms and LMS.

Reality

- Students in hard-to-reach territories of Ontario lack equitable access to hardware, software and especially bandwidth which will facilitate uninterrupted end-to-end online learning utilizing various educational technology platforms and LMS.
- The limited solutions which exist are less equitable, too costly, not scalable nor sustainable to be used in real-world and at large scale.

Consequences

- Hard-to-reach students in Ontario, with limited or no Internet connectivity, are unable to engage in and successfully complete online learning courses, offered by CN's partner course provides.

Proposed Solution

Based on the problem statement identified in the previous section, the following solution is proposed as a potentially viable solution:

- Identify a viable technology framework which will allow hard-to-reach students in Ontario, with limited or no Internet connectivity, to engage in and successfully complete online learning courses, offered by CN's partner course providers, in an offline learning environment.
- Create a scalable and sustainable workflow which will allow the creation, setup, distribution, and support of online courses offered by CN's partner course providers, in an offline learning environment.
- Design a robust technology architecture utilizing free and open source software (FOSS) which will require minimum setup, maintenance and equipment to run.
- Generate comprehensive documentation which will allow CN to utilize the solution in the real-world.

Prototype - Moodle Unplugged Framework

As a potential solution, the "Moodle Unplugged" technology framework was created allowing full functionality (with the exception of email, discussions and chat due to lack of Internet connectivity) of the Moodle LMS in an offline environment.

Technology architecture

- Moodle v3.11.2
    - o Interactive Content – H5P https://moodle.org/plugins/mod_hvp

    - H5P content types https://moodle.org/plugins/qformat_h5p
    - Enrolment key based self-registration https://moodle.org/plugins/auth_enrolkey/versions
    - PDF Document https://moodle.org/plugins/gradeexport_pdf
- XAMPP v7.3.11
    - XAMPP Control Panel v3.2.4
    - Apache 2.4.41
    - MariaDB 10.4.8
    - PHP 7.3.11 (VC15 X86 64bit thread safe) + PEAR
    - phpMyAdmin 4.9.1

More details on the technology architecture and implementation of Moodle Unplugged can be found in the learners', instructors' and administrators instructions.

Course Pack Creation and Distribution Workflow

The course creation and distribution workflow take into consideration four key stakeholders as shown in Figure 2.

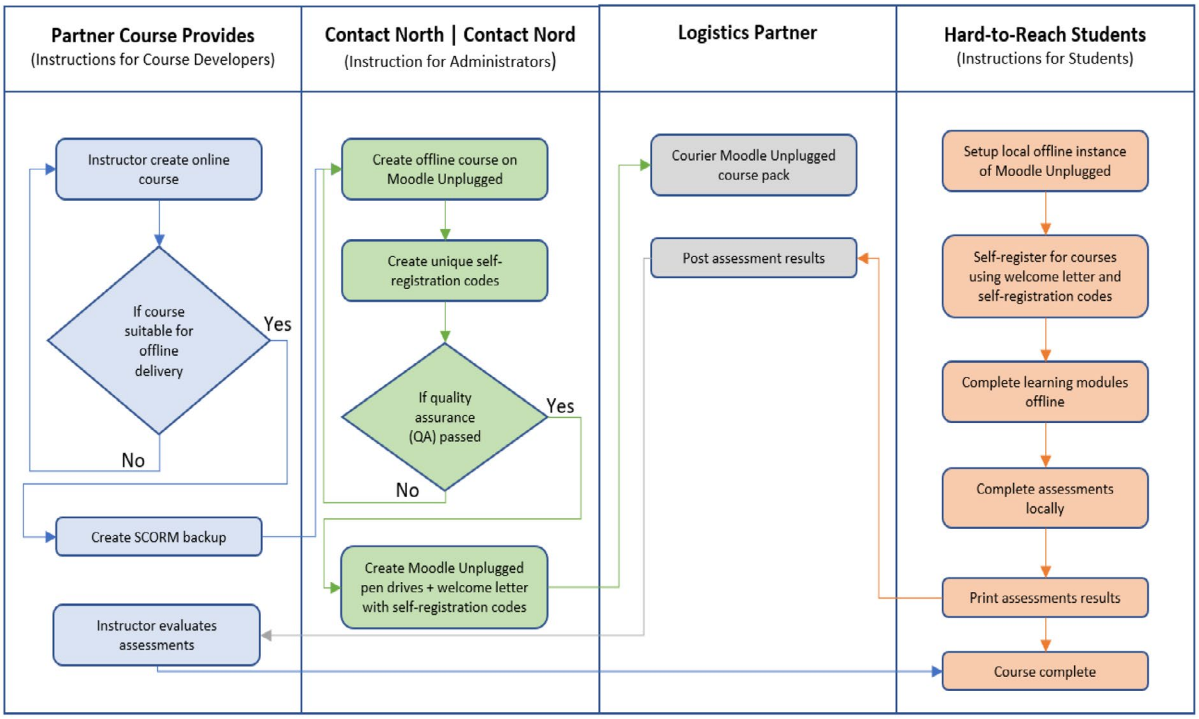
Partner Course Provides
(Instructions for Course Developers)
Contact North | Contact Nord
(Instruction for Administrators)
Logistics Partner
Hard-to-Reach Students
(Instructions for Students)
Instructor create online course
If course suitable for offline delivery
Yes
No
Create SCORM backup
Instructor evaluates assessments
Create offline course on Moodle Unplugged
Create unique self-registration codes
If quality assurance (QA) passed
Yes
No
Create Moodle Unplugged pen drives + welcome letter with self-registration codes
Courier Moodle Unplugged course pack
Post assessment results
Setup local offline instance of Moodle Unplugged
Self-register for courses using welcome letter and self-registration codes
Complete learning modules offline
Complete assessments locally
Print assessments results
Course complete

Figure 2 Moodle Unplugged course creation and distribution workflow.

Moodle Unplugged SWOT

Strengths

- Full functionality of the Moodle LMS in an offline environment (with the exception of email, discussions and chat due to lack of Internet connectivity)

- Students having almost the same learning experience as others in an online environment
- FOSS architectuture
- Workaround for completing quizess and submissions offline
- Ability to run courses created on different LMS platforms under common protocols
- Detailed step-by-step setup of Moodle Unplugged term version for administrators
- Step-by-step DIY setup of the system by student
- Distribution via courier and relatively inexpensive USB storage devices
- Relative low cost of implementation and economies of scale

Weaknesses

- Only compatible with Windows PC running Windows 7 or higher
- Discussion forums are inactive in offline mode
- E-mail communications are inactive
- Copyright clearance for offline resources such as YouTube videos, webpages etc.
- All the multimedia components used in the course need to be local versions uploaded onto the course rather than links to online resources such as embedded YouTube videos.
- H5P limitations
    - One of the key features of H5P is its ability to augment activities and learning interaction on top of videos, slides and other multimedia.
    - Most H5P interactions are built by superimposing learning activities on YouTube videos. This works fine in an online environment where bandwidth is ample.
    - In an offline environment, these videos don't play as they are directly streamed from YouTube. I.e. No Internet, no YouTube.
    - The design of H5P interactions for offline environment
        - The videos (or any other media) must be available locally and used as uploads (not links) when creating an H5P interaction
        - If using YouTube videos, the video must be downloaded locally and uploaded when creating the H5P interaction
        - By default, YouTube copyright doesn't allow local downloads.
        - If these videos are to be used as local versions, CN or the institution needs to clear copyright with the owner of the video before using in the H5P interaction
        - Alternatively, the institution could buy videos for use in this manner

    - Alternatively, the institution can use OER with CC BY license for this purpose; or
    - The institution can recreate these videos for use with H5P

Opportunities

- Wider adoption in remote parts of Canada beyond the province of Ontario
- Considerable impact if extended to rural communities where Internet penetration is minimum or non-existing
- Possibility of creating a lighter version for tablet computers and Chromebooks

Threats

- Increased availability and penetration of Internet
- Mobile app-based solutions
- Cost of hardware
- Cost of logistics
- Lack of technical expertise in administration and setup
- Lack of technical support for learners

ACKNOWLEDGEMENTS

I acknowledge the prompt and continued support provided by:

- Collaborative Media Group
- Literacy and Basic Skills (e-Channel) Coordinator, Contact North | Contact Nord
- Programmer/Analyst, Contact North | Contact Nord

APPENDIX A

A comparative review:

Note: information contained within the comparative review was the most current information available at the time of the review.

This review will compare the following categories:

- Cost
- Features and functionality
- Hardware/software requirements
- Set up
- Support

1. Kolibri

Basic Description:

Kolibri is an open-source educational platform specially designed to provide offline access to a wide range of quality, openly licensed educational resources in low-resource contexts like rural schools, refugee camps, orphanages, and in non-formal school programs.

https://kolibri.readthedocs.io/en/latest/

Kolibri is an adaptable set of open solutions specially developed to support learning for the half of the world without Internet access. Centered around an offline-first learning platform that runs on a variety of low-cost and legacy devices, the Kolibri Product Ecosystem includes a curricular tool, a library of open educational resources, and a toolkit of resources to support training and implementation in formal, informal, and non-formal learning environments.

https://learningequality.org/kolibri/

Features:

- Content Library (Kolibri Studio)
  - Kolibri Studio requires users to upload their content to the library to be able to use the content on a seeded device.
  - The Kolibri content library contains more than 80, ready to use, educational collections from educational organizations and creators.
  - See Public Catalog: https://studio.learningequality.org/en/channels/#/public
- Curricular Tool
  - Open-source, cloud-based, content studio also allows instructors to select and organize sets of resources, to meet their needs. Content must be added to the Kolibri studio to be seeded to server or device.
  - Add descriptions and custom exercises, and flag them as learner- or coach-facing, Kolibri Studio is the place to bundle,

remix, and describe materials to best meet your needs and those of the Kolibri users you are working with.

https://learningequality.org/kolibri/

Internet Dependency:

Initial set-up and subsequent upgrades require an Internet connection. After the initial set-up, devices running the Kolibri Learning Platform can find, and directly connect, with each other over a local Wi-Fi network.

Cost:

Varies depending on the device selected.

Free, open-source software. The Kolibri Learning Platform is designed to run on as many devices and operating systems as possible, leveraging existing legacy hardware or low-cost off-the-shelf devices. This includes Windows, Linux and macOS.

Hardware:

To run Kolibri, you will need the following hardware:

Power source: to supply your equipment such as server and client devices, can be using electricity, solar power, or generators. The power source selected should coincide with the length of instruction time required and can vary from one location to another. Devices can be charged once a day when used device battery life is sufficient.

Local network: to connect client devices to the server through the LAN (Local Area Network). A wired network can be used by connecting server and client devices to a router using ethernet cords. A router can be use WiFi to allow clients to connect wirelessly if available. The number of devices supported will depend on the router used.

https://drive.google.com/file/d/1ay1hLw1rNBGx3gloYHvyRxg5xf3QBS67/view

Server device(s): for example, a Raspberry Pi computer or laptop computer.

Depending on the device chosen for the server, you may require multiple servers to make Kolibri available to many classrooms and schools.

Important note: For best results, your server must always be connected to the network router using a cable.

Minimum hardware requirements to run Kolibri as a server:

- RAM: 1 GB recommended on Linux, 2GB or more on Windows devices
- Intel Core 3 CPU with 1 GHz recommended
- Hard drive space depends on the size of the channels you intend to import from Kolibri Studio or a local storage device.

- If you have a facility with less than 30 computers, a device as simple as a Raspberry Pi is known to work fine as a server.

Client device(s): for example, tablets, other computers, or mobile devices that are able to run a web browser. A web browser can be used to access Kolibri on the server using a local network connection. The selected client device must be able to run a supported browser.

Recommended browsers include either Google Chrome or Mozilla Firefox.

Note: Both server and client devices require a video card, sound card, and associated drivers so that content can be fully viewable. Server and client devices also require an updated browser to interact with Kolibri.

(https://drive.google.com/file/d/1ay1hLw1rNBGx3gloYHvyRxg5xf3QBS67/view)

Additional hardware configurations: https://kolibri.readthedocs.io/en/latest/faq.html#how-much-does-kolibri-cost-to-install

Software:

Operating Systems

Windows

- Supported: Windows 7, 8.1 and 10, with IE 11+, Chrome or Firefox

Debian/Ubuntu

- Debian 8 (Jessie), Debian 9 (Stretch), Ubuntu 16.04+ and up - anything that's not end-of-life. Ubuntu-based distributions have a number of flavors, for instance Xubuntu and Mint.

Raspberry Pi

- Kolibri is tested to work on Raspberry Pi Models 3, 3+, 4 and Zero W.
- Kolibri may also run on Raspberry Pi Model 2, but the platform has not been tested on that model, and it does not have on-board WiFi.

MacOS

- Kolibri macOS app has been tested on High Sierra (10.13), Mojave (10.14), and Catalina (10.15) macOS releases.

Python

- You can install Kolibri as a standard package from PyPi (works on Mac, Windows, and Linux) using this command:
  - Install for your current user.
  - pip install –user kolibri
  - Install system-wide (for all users).
  - sudo pip install kolibri

    - After Kolibri has been installed from PyPi, a new kolibri command should be available in your Terminal application.

  Download files for above operating systems: https://learningequality.org/download/

Initial Set-up:

You will only need to do the initial setup once when you first start Kolibri after the installation.

https://kolibri.readthedocs.io/en/latest/install/initial_setup.html

Set-up requires:

- Seeding a Device
    - Kolibri installers, updates, and content can be downloaded once to a device in an area that has an Internet connection
- Peer-to-Peer Distribution
    - That “seeded” device can then share new content and updates with other devices over an offline local network.
- Last Mile via “Sneakernet”
    - To reach the most remote communities, a device can be carried by foot to share installers, updates, and content with other devices over local networks.

https://learningequality.org/kolibri/

Quick Start guide: https://kolibri.readthedocs.io/en/latest/install/initial_setup.html#quick-start

Provisioning multiple servers: https://kolibri.readthedocs.io/en/latest/install/provision.html

Manual RaspberryPi setup: https://kolibri.readthedocs.io/en/latest/install/raspberry_pi_manual.html

Technical Support/Resources

https://kolibri.readthedocs.io/en/latest/manage/get_support.html#support_

https://kolibri.readthedocs.io/en/latest/contact.html

Admin documentation:

https://kolibri.readthedocs.io/en/latest/manage/

Instructor documentation:

https://kolibri.readthedocs.io/en/latest/coach/

Learner Documentation:

https://kolibri.readthedocs.io/en/latest/learn.html

General documentation:

https://learningequality.org/kolibri/

https://learningequality.org/documentation/

2. FreedomBox

Basic Description:

FreedomBox is made up of two things: a free and open-source software system and inexpensive hardware. The costs of computer processors and network bandwidth have both fallen to such low levels that hosting your own digital services is now affordable.

FreedomBox is a private server for non-experts: it lets you install and configure server applications with only a few clicks. It runs on cheap hardware of your choice, uses your Internet connection and power, and is under your control.

Features:

- File sharing
- VPN
- Chats/audio/video calls
- Proxy server
- Host wiki/blog/website
- Calendar and contact synching

https://wiki.debian.org/FreedomBox/Features

Internet Dependency:

WIFI can be used to provide Internet connectivity, or it can be used to share Internet connectivity with devices on the same network.

See the Networks manual page for instructions.

Cost:

Approximately $100CAD (69EUR) for the recommended Pioneer Kit

Hardware:

The Pioneer Edition FreedomBox Home Server Kits are the recommended pre-installed hardware for all users who don't wish to build their own FreedomBox by choosing the right components, downloading the image and preparing an SD card with FreedomBox.

The kit includes all the hardware needed for launching a FreedomBox home server on an Olimex A20-OLinuXino-LIME2 board.

Build your own hardware instructions.

Software:

FreedomBox is Free and Open-Source Software and an official part of Debian, a well-established GNU/Linux distribution.

Operating Systems:

Linux based on Debian.

Debian, the universal operating system, supports a much wider range on hardware. After installing Debian, see the manual page for installing FreedomBox on Debian.

Initial Set-up:

FreedomBox offers a pre-built images that can be used for set-up, allowing users to determine if they would rather use the pre-built image or create their own using the documentation available.

https://wiki.debian.org/FreedomBox/Setup#:~:text=FreedomBox%20Setup%20is%20a%20Debian,accounts%2C%20installing%20essential%20packages%20etc.

Set-up requirements:

Set-up requirements depend on if you are using a pre-built image or starting from scratch.

Set-up requirements are available from the following sources:

- Git repository https://salsa.debian.org/freedombox-team/freedombox-setup
- building a Debian package from source files https://salsa.debian.org/freedombox-team/freedombox-setup/-/tree/master

https://wiki.debian.org/FreedomBox/Setup

Technical Support/Resources:

https://freedombox.org/#community
https://wiki.debian.org/FreedomBox/Manual/GettingHelp?action=show&redirect=FreedomBox%2FSupport

https://www.zdnet.com/article/put-the-Internet-back-under-your-control-with-the-freedombox/

Home Server: https://wiki.debian.org/FreedomBox

Build your own: https://freedombox.org/download/