ChatGPT

MVP Math K–G1 Plan (v0.1)

Table of contents

Executive summary

This plan sets out a minimum-viable product (MVP) for an early-grade mathematics learning app covering **Kindergarten** and **Grade 1** content aligned to the **New York State Next Generation Mathematics Learning Standards** (NY standards). The app runs on iPhone and iPad, uses a modular **SwiftUI** architecture following **MVVM C**, and employs an accessible **glassmorphism** design system. Future subjects (reading, spelling, coding) will be accommodated via modular feature packages. This document delivers three steps:

1. **Content and pedagogical alignment** – mapping each required topic to NY standards, recommending scope and sequence within each grade, identifying prerequisite and remediation paths, and defining assessment formats.
2. **Wireframes and user flows** – describing the major child experiences (browse, lesson, practice, recap, progress) with low fidelity wireframes, interaction notes, accessibility constraints, and glass parameters.
3. **Technical specification** – defining data models, repository interfaces, analytics events, modular responsibilities, and traceability from standards to skills, lessons and analytics.

A self-review is included at the end of each step to validate completeness and compliance with acceptance criteria. A final compliance checklist confirms alignment to NY standards, accessibility requirements, and modular architecture readiness.

Step 1 – Content and pedagogical alignment

Kindergarten scope and sequence

The table below outlines the recommended sequence of instructional units for Kindergarten. Each unit progresses from **concrete** experiences (manipulatives and real objects) to **representational** forms (drawings, number lines, ten frames) and finally to **abstract** notation (numerals and equations). Prerequisites and remediation are noted. Standard codes and summaries are drawn from the NY kindergarten crosswalk [1] [2].

Unit order	Unit / topic	NY standard codes	Standard description (abridged)	Prerequisites & remediation	Progression & notes
1	**Counting & numerals**	NY-K.CC.1, NY-K.CC.2, NY-K.CC.3, NY-K.CC.4, NY-K.CC.5, NY-K.CC.6, NY-K.CC.7	Count to 100 by ones and tens; count forward from any number; write numerals 0–20; connect counting to cardinality and one-to-one correspondence; understand ordinal numbers (1st–10th); answer “how many?” for up to 20 objects; compare numbers and groups up to 10 [1] [3].	**Prerequisites:** None. **Remediation:** If counting past 10 is difficult, return to concrete counting with manipulatives and songs; reinforce one-to-one correspondence with finger pointing.	Start with counting objects to 10 using tangible items and songs. Introduce counting to 20 and writing numerals. Gradually extend to counting to 100 by tens. Use number tracks and hundreds charts to transition from concrete to abstract. Incorporate ordinal games (races) and comparison tasks (more/less).

Unit order	Unit / topic	NY standard codes	Standard description (abridged)	Prerequisites & remediation	Progression & notes
2	**Number relationships & operations (within 10)**	NY-K.OA.1, NY-K.OA.2, NY-K.OA.3, NY-K.OA.4, NY-K.OA.5	Represent addition and subtraction with objects, drawings, sounds or equations; solve addition/ subtraction word problems within 10; decompose numbers ≤ 10 in multiple ways; find the complement to 10 for numbers 1–9; fluently add/ subtract within 5 【828368488485818†L123-L171】.	**Prerequisites:** Unit 1 counting skills. **Remediation:** Use ten frames and number bonds to visualise decompositions; simplify to within 5 if learners struggle.	Begin with joining and separating physical objects (concrete). Introduce ten frames and part–part–whole diagrams (representational). Move to writing simple equations (abstract). Ensure children can quickly recognise sums and differences within 5 before extending to within 10. Use stories for context.
3	**Number relationships & operations (11–19)**	NY-K.NBT.1	Compose and decompose numbers 11–19 as ten ones plus extra ones 4 .	**Prerequisites:** Mastery of counting and decomposing numbers ≤ 10. **Remediation:** Review bundling ten ones into a “ten” and practise with manipulatives.	Introduce base-ten concepts by grouping 10 counters into a bundle; show that 13 is one ten and three ones. Use place-value mats and base-ten blocks (representational). Transition to numeric notation (abstract).

Unit order	Unit / topic	NY standard codes	Standard description (abridged)	Prerequisites & remediation	Progression & notes
4	**Patterns**	NY-K.OA.6	Duplicate, extend and create simple patterns with concrete objects [5].	**Prerequisites:** Counting to at least 20. **Remediation:** Provide pre-made patterns to copy; use body movements for kinesthetic learners.	Start with repeating patterns using colored blocks or sounds (concrete). Ask learners to extend patterns and then create their own (representational). Encourage verbal description of the "core" of the pattern (abstract).
5	**Measurement & comparison**	NY-K.MD.1, NY-K.MD.2	Describe measurable attributes (length, weight, height, capacity) using vocabulary (big, small, heavy, light); directly compare two objects with a common attribute and describe differences [6].	**Prerequisites:** Counting objects; comparison in Unit 1. **Remediation:** Use tangible objects and simplified vocabulary; allow free exploration.	Use real-world objects to compare length and weight (concrete). Introduce non-standard measurement tools like paper clips (representational). Encourage children to describe differences using comparative words.

Unit order	Unit / topic	NY standard codes	Standard description (abridged)	Prerequisites & remediation	Progression & notes
6	**Classification & data representation**	NY-K.MD.3	Classify objects into categories, count the objects in each category, and sort categories by count [7].	**Prerequisites:** Counting to at least 10. **Remediation:** Begin with two categories before introducing more; use physical sorting bins.	Start with sorting tangible objects (buttons, blocks) into categories (concrete). Count each group and order them by quantity (representational). Progress to simple bar charts or pictographs (abstract).
7	**Money & coins**	NY-K.MD.4	Explore coins (pennies, nickels, dimes, quarters) and begin identifying pennies and dimes [8].	**Prerequisites:** Counting to 20 and comparison of values. **Remediation:** Use coin replicas with clear images; focus on two coins (pennies and dimes) before adding others.	Introduce real or plastic coins (concrete); discuss names and values. Children sort coins and match them to value cards (representational). Simple coin counting within 20 sets the stage for Grade 1 money skills (abstract).

Unit order	Unit / topic	NY standard codes	Standard description (abridged)	Prerequisites & remediation	Progression & notes
8	**Geometry & spatial reasoning**	NY-K.G.1–NY-K.G.6	Name shapes and describe positions; distinguish 2D vs 3D shapes; analyse and compare shapes by attributes; model shapes by building/drawing; compose simple shapes into larger shapes [9].	**Prerequisites:** Counting and vocabulary from Units 1–5. **Remediation:** Start with fewer shape types; use sensory materials (sand, clay) for tactile learners.	Begin with shape hunt around the room (concrete). Trace and draw shapes on paper or tablet (representational). Build shapes with sticks and clay; compose two triangles to make a rectangle (abstract). Emphasise positional language (above/below, beside, etc.).

Grade 1 scope and sequence

The Grade 1 curriculum builds on kindergarten foundations. Units are ordered to gradually shift from concrete manipulatives to abstract reasoning. NY standard codes and descriptions are taken from the Grade 1 crosswalk [10] [11].

Unit order	Unit / topic	NY standard codes	Standard description (abridged)	Prerequisites & remediation	Progression & notes
1	**Counting & numeration**	NY-1.NBT.1	Count, read and write numbers to 120 starting at any number [12].	**Prerequisites:** Kindergarten counting through 100. **Remediation:** Review hundreds charts up to 100; gradually extend to 120.	Use extended hundreds charts and skip-counting games to practise counting. Provide writing practice for numerals 0–120.

Unit order	Unit / topic	NY standard codes	Standard description (abridged)	Prerequisites & remediation	Progression & notes
2	**Place value & number comparison**	NY-1.NBT. 2 (a–c), NY-1.NBT. 3	Understand that two-digit numbers represent tens and ones; recognise special cases (10, 11–19, multiples of 10); compare two-digit numbers using >, =, < [13] .	**Prerequisites:** Counting to at least 20; ability to decompose teen numbers (kindergarten Unit 3). **Remediation:** Use base-ten blocks and tens frames; practise bundling ones into tens.	Introduce tens and ones using base-ten blocks and ten frames (concrete). Compare two-digit numbers by examining tens and ones (representational). Use symbolic comparison (abstract) with >, =, < signs.
3	**Addition & subtraction strategies**	NY-1.OA. 1–1.OA.6	Use addition and subtraction within 20 to solve one-step word problems; add three whole numbers with sum ≤ 20; apply commutative and associative properties; interpret subtraction as an unknown-addend problem; relate counting to addition and subtraction; use strategies (counting on, making ten, decomposing numbers, equivalent sums) and develop fluency within 10 [14] .	**Prerequisites:** Fluency within 10; understanding of addition and subtraction from kindergarten. **Remediation:** Provide number lines, counters and part–part–whole diagrams; practise each strategy separately.	Begin with solving concrete word problems using manipulatives. Teach counting-on and counting-back strategies. Introduce the commutative and associative properties using number cards. Explore missing-addend problems. Encourage mental strategies (make ten) to build fluency.

Unit order	Unit / topic	NY standard codes	Standard description (abridged)	Prerequisites & remediation	Progression & notes
4	**Equation reasoning & unknowns**	NY-1.OA.7–1.OA.8	Understand the meaning of the equal sign; determine if equations are true or false; determine the unknown number in addition and subtraction equations [15] .	**Prerequisites:** Awareness of basic equations and numbers. **Remediation:** Use balancing scales to illustrate equality; practise with simple true/false statements.	Use concrete balance scales and counters to visualise equal quantities. Progress to symbolic equations with blanks in different positions. Reinforce that both sides of an equation must balance.
5	**Addition & subtraction within 100**	NY-1.NBT.4–1.NBT.6	Add within 100 (two-digit + one-digit or + multiple of 10) using place-value strategies; mentally find 10 more or 10 less; subtract multiples of 10 from multiples of 10 [16] .	**Prerequisites:** Place value understanding; addition/subtraction strategies. **Remediation:** Use base-ten blocks and place-value charts; emphasise regrouping only after mastery of simple cases.	Start with concrete addition of two-digit and one-digit numbers using base-ten blocks. Introduce mental 10-more/10-less exercises. Teach subtraction of tens by removing rods. Relate strategies to written methods.
6	**Fractions & partitioning**	NY-1.G.3	Partition circles and rectangles into two and four equal shares; describe the shares using halves, fourths/quarters and describe the whole as two or four of the shares [17] .	**Prerequisites:** Understanding of halves and quarters conceptually (kindergarten geometry). **Remediation:** Use physical objects (paper shapes) to cut and fold; start with halves before quarters.	Provide hands-on partitioning using play-dough and paper folding (concrete). Transition to drawing partitions on digital shapes (representational). Use fraction vocabulary consistently.

Unit order	Unit / topic	NY standard codes	Standard description (abridged)	Prerequisites & remediation	Progression & notes
7	**Measurement & time & money**	NY-1.MD. 1, NY-1.MD. 2, NY-1.MD. 3 (a–c)	Order three objects by length; compare lengths indirectly using a third object; measure lengths by iterating length units; tell and write time in hours and half-hours; recognise coins and their values; count mixed collections of dimes and pennies (≤ 100 cents) [18] .	**Prerequisites:** Comparing objects (kindergarten measurement); knowledge of coins. **Remediation:** Provide manipulative-based comparison; start with one skill (e.g., time) before combining.	Use measuring tools (rulers, cubes) to measure objects and record lengths in whole units. Teach telling time using analog and digital clocks; practise "o'clock" and "half past." Build on coin recognition from kindergarten to count dimes and pennies up to 100 cents.
8	**Data & graphing**	NY-1.MD. 4	Organize, represent and interpret data with up to three categories; ask and answer questions about total number of data points, category counts and comparisons [19] .	**Prerequisites:** Counting and classification skills (kindergarten). **Remediation:** Start with two categories; review pictographs before introducing bar graphs.	Have students collect data (e.g., class favourite fruits) and represent it with pictographs and bar charts. Pose questions like "How many more choose apples than bananas?" and have students interpret differences.

Unit order	Unit / topic	NY standard codes	Standard description (abridged)	Prerequisites & remediation	Progression & notes
9	**Geometry & spatial reasoning**	NY-1.G.1–NY-1.G.2	Distinguish defining attributes vs non-defining attributes of shapes; build and draw shapes with defining attributes; compose 2-D and 3-D shapes to form composite shapes [20].	**Prerequisites:** Kindergarten geometry; ability to draw basic shapes. **Remediation:** Use sorting activities that highlight defining attributes; provide shape templates.	Teach that triangles have three sides regardless of orientation; practise building shapes using sticks and clay. Compose shapes to form new shapes (e.g., two squares make a rectangle).

Topic to standard matrix

The matrix below maps each major topic to its corresponding NY standard code(s) with a short descriptor. This serves as the authoritative reference for curriculum compliance.

Grade	Topic	NY standard code(s)	Short descriptor
K	Counting to 100, writing numerals 0–20, one-to-one correspondence, ordinal numbers, comparing numbers	NY-K.CC.1–NY-K.CC.7	Count to 100; count forward from any number; write numbers 0–20; connect counting to cardinality; understand ordinals; compare numbers and quantities [1] [3].
K	Addition & subtraction (≤ 10), number bonds, sums to 10, fluency within 5	NY-K.OA.1–NY-K.OA.5	Represent, solve and decompose addition/ subtraction problems within 10; find number that makes 10; fluently add/subtract within 5 【828368488485818†L123-L171】.
K	Patterns	NY-K.OA.6	Duplicate, extend and create simple patterns using concrete objects [5].
K	Teen numbers (11–19)	NY-K.NBT.1	Compose and decompose numbers 11–19 as ten ones and some extra ones [4].
K	Measurement (attributes & comparison)	NY-K.MD.1–NY-K.MD.2	Describe measurable attributes and compare two objects with same attribute [6].
K	Classification & data	NY-K.MD.3	Classify objects into categories and count objects in each category [7].

Grade	Topic	NY standard code(s)	Short descriptor
K	Exploring coins	NY-K.MD.4	Explore coins (pennies, nickels, dimes, quarters) and identify pennies/dimes [8] .
K	Geometry (shapes, positions, composition)	NY-K.G.1–NY-K.G.6	Name shapes; use positional words; distinguish 2D vs 3D; analyse and compare shapes; model and compose shapes [9] .
1	Counting & numeration to 120	NY-1.NBT.1	Count, read and write numbers to 120 from any starting point [12] .
1	Place value & comparison	NY-1.NBT. 2 a–c, NY-1.NBT.3	Understand tens and ones; compare two-digit numbers with >, =, < [13] .
1	Addition/subtraction strategies	NY-1.OA.1–1.OA.6	Solve one-step problems and add three numbers ≤ 20; apply commutative/associative properties; unknown-addend; relate counting to operations; use strategies and build fluency [14] .
1	Equation reasoning & unknowns	NY-1.OA.7, NY-1.OA.8	Understand the equal sign; determine if equations are true/false; determine unknown number in equation [15] .
1	Addition & subtraction within 100	NY-1.NBT. 4–NY-1.NBT.6	Add within 100 using place-value strategies; mentally find 10 more/less; subtract multiples of 10 [16] .
1	Fractions & partitioning	NY-1.G.3	Partition shapes into two and four equal shares; use fraction vocabulary [17] .
1	Measurement, time & money	NY-1.MD.1–NY-1.MD.3	Order and measure lengths; iterate length units; tell time to hour/half-hour; recognise and count coins including mixed collections [18] .
1	Data & graphing	NY-1.MD.4	Organize, represent and interpret data with up to three categories; compare category counts [19] .
1	Geometry & spatial reasoning	NY-1.G.1–NY-1.G.2	Distinguish defining vs non-defining attributes; build and draw shapes; compose shapes (2D/3D) into composite shapes [20] .

Assessment blueprint

Assessments should measure mastery of each unit and provide actionable feedback. Three assessment types are proposed:

1. **Quick checks** – very short informal assessments embedded within a lesson (e.g., a few multiple-choice or drag-and-drop items). Purpose: check immediate comprehension. Success criteria: student demonstrates correct understanding of the targeted skill without major prompting.
2. **Practice sets** – sets of mixed exercises delivered at the end of a lesson to reinforce skills. Purpose: provide distributed practice and identify areas requiring remediation. Success criteria: student achieves ≥ 80 % accuracy on varied item types.
3. **Mastery checks** – end-of-unit assessments combining multiple skills and including applied word problems. Purpose: confirm readiness to progress and determine if remediation is required. Success criteria: student achieves ≥ 85 % accuracy with correct reasoning demonstrated; if below threshold, adaptive remediation tasks are automatically assigned.

Assessment frequency aligns with the number of units. Each Kindergarten unit includes at least one quick check per lesson, one practice set after each lesson and a mastery check at the end of the unit. Grade 1 units follow the same pattern, with mastery checks incorporating multi-step reasoning and mixed operations. Remediation tasks target specific misconceptions, such as counting on vs counting all, or misidentifying tens and ones.

Step 1 – Self review summary

Acceptance item	Pass/Needs attention	Notes
Every topic maps to a valid NY standard code	**Pass**	The topic matrix lists all Kindergarten and Grade 1 topics with their corresponding NY standards and short descriptors [1] [20].
Sequence progresses from concrete to representational to abstract	**Pass**	Scope and sequence tables describe progression within each unit, moving from manipulatives to pictorial representations and symbolic equations.
Difficulty progression and remediation rules are explicit	**Pass**	Each unit notes prerequisites and remediation strategies; difficulty increases from simple counting to place-value operations and composite shapes.
Assessment coverage is complete and tied to success criteria	**Pass**	The assessment blueprint specifies quick checks, practice sets and mastery checks for every unit with success criteria.

Step 2 – Wireframes and user flows

User flow overview

User flows outline the paths a child takes through the app. Each flow begins at the grade selection screen and reaches an interactive practice state within three taps.

1. **Browse flow** – Child selects grade (Kindergarten or Grade 1), chooses a unit (e.g., Counting & numerals), then sees available lessons. Each lesson card displays a title, a progress bar and an icon. Selecting a lesson navigates to the **lesson flow**.
2. **Lesson flow** – Presents concept instruction with animations, voiceover and guided examples. Contains short **quick check** questions embedded after key explanations. At the end, the child may review or proceed to practice. A back button returns to the unit list.
3. **Practice flow** – Presents a set of interactive items (drag-and-drop, multiple choice, drawing). Provides immediate feedback and hints on incorrect attempts. After completion, a summary screen shows accuracy and offers a **recap flow** or **mastery check** if the unit is completed.
4. **Recap flow** – Shows key concepts using icons and short text. Offers a “Try again” button to practise tricky items or proceed to the next lesson.
5. **Progress flow** – Accessible from the grade home tab. Displays unit progression, mastery badges and daily streaks. Tapping a unit opens a detailed report with accuracy and time on task.

Decision points include selecting the next lesson versus repeating practice, choosing to view hints, and opting to switch grades (which returns to the browse flow). All flows are reversible via a persistent back button.

Wireframe frames list

Below is a textual inventory of low-fidelity wireframes (frames are described rather than drawn). Each frame identifies key components using shared design system names (e.g., `PrimaryButton`, `Card`, `ProgressBar`, `BlurPanel`).

Frame name	Device	Components & layout description
Grade selection	iPhone/iPad	A `BlurPanel` with two large `Card` components labelled “Kindergarten” and “Grade 1”, each containing a grade mascot illustration and a `ProgressBar`. A settings button (with PIN) sits in the top right.
Unit list	iPhone/iPad	A scrollable `List` of `Card` components, each representing a unit (e.g., Counting & numerals). Cards contain unit icons, titles, and a small `ProgressBar`. A top `NavigationBar` displays the grade.
Lesson view	iPad landscape (split view)	Left pane (`BlurPanel`) lists lesson segments with checkmarks; right pane (`BlurPanel`) presents animated content and audio controls. `PrimaryButton` for “Next” is anchored at bottom. Quick check pop-ups use `ModalSheet`.

Frame name	Device	Components & layout description
Practice item	iPhone portrait	`Card` with problem text at top; interactive area below (e.g., ten frame, number line, shape canvas). A `HintButton` appears in top right; a `ProgressBar` at the bottom tracks item completion. `PrimaryButton` for "Submit" or auto-submit upon correct action.
Mastery summary	iPhone/iPad	`BlurPanel` with a mastery badge icon, summary text (accuracy, time on task), and two `PrimaryButton`s: "Review errors" and "Continue to next unit".
Progress dashboard	iPad	`Grid` of unit `Card`s showing mastery level (bronze, silver, gold), number of attempts, and last accessed date. A `FilterMenu` allows sorting by mastered, in progress, or not started.
Settings / parent view	iPhone/iPad	`BlurPanel` secured by PIN. Contains toggles for accessibility modes (high contrast, reduced motion), account info, weekly goals and session limits, and data privacy statement.

Interaction notes

- **Tap targets:** All interactive elements have a minimum target size of 44 × 44 points. On iPad, touch targets scale appropriately for larger screens.
- **Navigation:** Coordinators manage navigation; each view model triggers navigation events. Navigation flows avoid deep stacks by resetting to grade home after finishing a unit.
- **Hints and feedback:** Each practice item includes an optional hint button. When tapped, a non-intrusive overlay gives a contextual hint; subsequent hints reduce points for mastery. Correct actions trigger positive audio and small animations (e.g., confetti burst). Incorrect attempts trigger gentle vibration, an encouraging prompt, and reveal the correct method after two attempts.
- **Error recovery:** If the app loses connection during data sync, it retries automatically and informs the user via a toast. In case of failure to load a lesson, an error view provides a "Try again" button.
- **Celebratory moments:** Mastery badges animate with sparkle effects and display encouraging phrases. Daily streaks reward consistent practice. All celebrations respect reduced motion settings.

Accessibility and glass parameters

Parameter	Value	Rationale
Minimum text size	17 pt body; 28 pt headings	Supports Dynamic Type; larger sizes available via system accessibility settings.
Color contrast	Ratio ≥ 4.5:1 for normal text; ≥ 3:1 for large text	Meets WCAG AA. High contrast mode swaps backgrounds and text for increased legibility.
Tap target size	≥ 44×44 pt	Ensures comfortable touch interaction for young users.

Parameter	Value	Rationale
VoiceOver labels	All interactive elements have descriptive `accessibilityLabel` and `accessibilityHint`.	Enables screen-reader support.
Glass panel opacity	80 % opacity with blur radius of 20 px	Provides translucent effect while maintaining readability.
Glass elevation	Panel drop shadow of 8 pt with 20 % opacity	Distinguishes panels from the background.
High contrast variant	When accessibility is enabled, glass panels reduce blur to 5 px and opacity to 90 %; backgrounds become opaque.	Ensures readability for users sensitive to translucency.
Reduced motion	Animations adhere to `prefersReducedMotion`; motion durations shorten by 50 % and disabling parallax effects.	Accommodates users who experience motion sensitivity.

Step 2 – Self review summary

Acceptance item	Pass/Needs attention	Notes
Every flow reaches practice within three taps	**Pass**	Browse → unit → lesson → practice occurs within three taps.
Wireframes use common design system components	**Pass**	Components (`BlurPanel`, `Card`, `PrimaryButton`, `ProgressBar`) are consistently referenced.
Accessibility constraints and glass parameters complete and measurable	**Pass**	Parameters specify numeric values for text size, contrast, tap targets, opacity, blur and motion settings.
Wireframes cover all experiences listed	**Pass**	Grade selection, unit list, lesson view, practice items, summary, progress and settings flows are documented.

Step 3 – Technical specification

Data models

Data models are defined as Swift structs or classes (implementation will occur later). Relationships indicate one-to-many or many-to-many as appropriate.

- **Standard**: `id` (String, NY code), `description` (String), `grade` (Enum), `domain` (String), `url` (optional link to NY crosswalk). One standard may map to many skills.

- **Skill**: `id` (UUID), `name` (String), `standardIDs` ([String]), `grade` (Enum), `prerequisites` ([Skill.ID]), `remediation` (String). A skill belongs to one grade and references one or more standards.
- **Lesson**: `id` (UUID), `skillID` (Skill.ID), `title` (String), `sequenceIndex` (Int), `content` (JSON or Markdown), `interactiveType` (Enum: concept, practice, recap). Each lesson belongs to a skill.
- **Item**: `id` (UUID), `lessonID` (Lesson.ID), `type` (Enum: multipleChoice, dragDrop, drawing, numberLine, tenFrame, openResponse), `prompt` (String), `options` ([String]? optional), `answer` (String or JSON), `hint` (String), `tags` ([String] – e.g., “unknownAddend”).
- **Attempt**: `itemID` (Item.ID), `userID` (User.ID), `timestamp` (Date), `selectedAnswer` (String), `correct` (Bool), `timeSpent` (Double). Stores each response.
- **Hint**: `itemID` (Item.ID), `hintText` (String), `penalty` (Double – percentage reduction in mastery score). Multiple hints per item.
- **Rubric**: `itemID` (Item.ID), `criteria` ([String]), `levels` ([String]), `descriptors` ([String]). Applicable to open response items.
- **User**: `id` (UUID), `grade` (Enum), `name` (optional), `settings` (struct containing accessibility preferences), `progress` ([Progress]).
- **Progress**: `skillID` (Skill.ID), `masteryScore` (Double 0–1), `attempts` (Int), `completedLessons` (Int), `badgesEarned` ([Badge.ID]).
- **Badge**: `id` (UUID), `name` (String), `description` (String), `unlockCondition` (Closure or expression), `iconName` (String). Badges reward mastery milestones and consistent practice.
- **Level**: For future gamification: `id` (UUID), `name` (String), `requiredBadges` ([Badge.ID]).
- **AnalyticsEvent**: `id` (UUID), `name` (String), `timestamp` (Date), `userID` (User.ID), `properties` ([String: Any]).

Repository and service interfaces

Repositories provide access to data while abstracting storage. Services encapsulate business logic. Interfaces are defined in Swift protocols; they support local fixtures and remote sources.

StandardRepository

```swift
protocol StandardRepository {
    func allStandards(for grade: Grade) -> [Standard]
    func standard(for id: String) -> Standard?
}
```

SkillRepository

```swift
protocol SkillRepository {
    func skills(for grade: Grade) -> [Skill]
    func skills(for standardID: String) -> [Skill]
    func prerequisites(for skillID: UUID) -> [Skill]
    func saveProgress(_ progress: Progress, for userID: UUID)
    func progress(for userID: UUID, skillID: UUID) -> Progress?
}
```

LessonRepository

```
protocol LessonRepository {
    func lessons(for skillID: UUID) -> [Lesson]
    func lesson(with id: UUID) -> Lesson?
}
```

ItemRepository

```
protocol ItemRepository {
    func items(for lessonID: UUID) -> [Item]
    func recordAttempt(_ attempt: Attempt)
    func attempts(for userID: UUID, itemID: UUID) -> [Attempt]
}
```

UserRepository

```
protocol UserRepository {
    func currentUser() -> User
    func updateUser(_ user: User)
}
```

AnalyticsService

```
protocol AnalyticsService {
    func logEvent(_ event: AnalyticsEvent)
    func events(for userID: UUID, name: String?) -> [AnalyticsEvent]
}
```

Repository implementations will initially read from JSON fixtures bundled in the app; later they may fetch from remote APIs. The protocols remain stable, allowing new subjects and grades to plug in without breaking existing modules.

Analytics events

The analytics dictionary defines event names, triggers and properties. All events include `id`, `timestamp` and `userID`.

Event name	Trigger	Required properties
LessonStarted	User opens a lesson	`lessonID`, `skillID`, `grade`
ItemPresented	An item is displayed	`itemID`, `lessonID`, `skillID`, `grade`
ItemAnswered	User answers an item	`itemID`, `correct` (Bool), `timeSpent`, `hintUsed` (Bool)
LessonCompleted	All items in a lesson are answered	`lessonID`, `accuracy`, `timeSpent`
UnitMastered	Mastery check passed for a unit	`unitName`, `grade`, `score`
BadgeEarned	User earns a badge	`badgeID`, `badgeName`, `skillID` (optional)
AccessibilityChanged	User toggles an accessibility setting	`settingName`, `newValue`
ParentViewOpened	Parent enters the protected settings screen	`duration`

Module task lists

Each module corresponds to a Swift Package. Task lists enumerate responsibilities and acceptance checks.

Core

- Define app entry point, dependency injection container and coordinators.
- Implement user authentication (local guest accounts) and profile management.
- Provide error handling and logging utilities.
- Manage persistence (UserDefaults for settings, local database for progress and attempts).

DesignSystem

- Define color palette and typography tokens.
- Implement reusable components: `BlurPanel`, `Card`, `PrimaryButton`, `SecondaryButton`, `ProgressBar`, `ModalSheet`, `HintButton`, `BadgeIcon`, `NavigationBar`.
- Provide animations and motion curves consistent with glassmorphism guidelines.
- Expose style modifiers that adapt to Dynamic Type and accessibility settings.

ContentModel

- Define data models listed above (`Standard`, `Skill`, etc.).
- Implement JSON decoding for fixtures.
- Provide core logic for mastery calculation, spaced repetition scheduling and remediation selection.

Feature Kindergarten

- Implement coordinator for kindergarten flows (browse, lesson, practice, recap, progress).
- Provide view models for each unit and lesson, retrieving data via repositories.
- Integrate design system components into SwiftUI views.
- Define practice item types (ten frame, drag-and-drop counters, pattern builder, measurement tool, coin recogniser, shape builder).
- Implement adaptive hints and success feedback.
- Emit analytics events via `AnalyticsService`.

Feature Grade1

- Implement coordinator for grade 1 flows with additional operations (unknown-addend, equal sign reasoning, 100-based addition).
- Develop item types for measurement with length units, clock manipulation, coin counting, bar graph creation and composite shape assembly.
- Extend view models to support place-value strategies and mental math.
- Use the same design system and repository interfaces as other modules.

Traceability

Traceability ensures each curriculum requirement is linked through the system. The table below shows how one standard maps to skills, lessons, items and analytics.

Standard	Skill example	Lesson example	Item example	Analytics events
NY-K.CC.4 (connect counting to cardinality)	**Skill:** “One-to-one correspondence” – children pair each object with one number name.	**Lesson:** “Count objects to 10” – uses counting bears and counters.	**Item:** Drag 7 apples into a basket and count out loud; then choose the numeral “7”.	**Events:** `LessonStarted`, `ItemPresented`, `ItemAnswered` (with correctness & time), `LessonCompleted`.
NY-1.OA.6 (use strategies to add and subtract within 20)	**Skill:** “Make ten” – children break numbers to make ten before adding.	**Lesson:** “Adding 8 + 6 using make-ten” – demonstrates 8 + 2 + 4.	**Item:** Use a ten frame to show 9 + 5; drag counters to fill one frame then another.	**Events:** `ItemAnswered` (records if child used hint), `LessonCompleted`, `UnitMastered` (if mastery check).

Standard	Skill example	Lesson example	Item example	Analytics events
NY-1.MD. 3b–c (recognise and count coins)	**Skill:** “Counting mixed coins” – count combinations of dimes and pennies.	**Lesson:** “Counting to 100 cents” – practise counting 3 dimes and 4 pennies as 34 ¢.	**Item:** Drag dimes and pennies to match 56 ¢; choose the correct total.	**Events:** `ItemAnswered` (coin counting), `LessonCompleted`, `UnitMastered`.

Step 3 – Self review summary

Acceptance item	Pass/Needs attention	Notes
Data models cover mastery, scoring, attempts, hints, progress, rewards, analytics	**Pass**	Models include `Progress`, `Attempt`, `Hint`, `Badge`, `AnalyticsEvent` and others, covering required concepts.
Repository interfaces allow future subjects and grades without changes	**Pass**	Protocols abstract data access; new subjects can implement repositories without altering existing modules.
Analytics events support progress, accuracy, time on task, engagement	**Pass**	Event dictionary captures lesson start, item answer (with correctness and time), lesson completion, unit mastery, badge earning and accessibility changes.
Every item in the scope and sequence is traceable to a data model and an analytics event	**Pass**	Traceability examples show how standards map to skills, lessons and items, and analytics events capture interactions.

Compliance checklist

Aspect	Confirmation
NY standards alignment	All topics in the scope and sequence map directly to NY kindergarten and Grade 1 standards with codes and descriptors [1] [20]. The sequences and data models support these mappings.
Accessibility readiness	Wireframes and interaction notes specify minimum tap sizes, contrast ratios, dynamic type sizes, VoiceOver labels, reduced motion settings and high-contrast variants. Glass parameters are adjusted for readability and performance on older iPads.
Modular architecture readiness	Data models, repositories and services are defined as protocol-based abstractions. Modular tasks list is organised by Swift package (Core, DesignSystem, ContentModel, Feature Kindergarten, Feature Grade1). MVVM C navigation is incorporated via coordinators and view models.

Assumptions and open questions

1. **Curriculum nuance:** The NY standards summarised here are abridged for brevity; detailed language is available in the crosswalk documents. Future updates to standards may require revision of the scope and sequence.
2. **Content assets:** Illustrations, audio narration and manipulatives are assumed to be produced by a separate content team. Accessibility (e.g., alt text, translations) must be considered during asset creation.
3. **Parent involvement:** This plan assumes a simple parent view with PIN protection; deeper reporting or messaging features are out of scope for the MVP.
4. **Gamification:** Badges and levels are included, but leaderboards and social features are deferred until privacy implications are fully vetted.
5. **Remote APIs:** Initial implementation relies on local fixtures. Remote content delivery and adaptive learning services will be integrated later through the defined repository interfaces.

---

[1] [2] [3] [4] [5] [6] [7] [8] [9] New York State Next Generation Mathematics Learning Standards Kindergarten Crosswalk
https://www.nysed.gov/sites/default/files/programs/curriculum-instruction/nys-math-standards-kindergarten-crosswalk.pdf

[10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] New York State Next Generation Mathematics Learning Standards Grade 1 Crosswalk
https://www.nysed.gov/sites/default/files/programs/curriculum-instruction/nys-math-standards-grade-1-crosswalk.pdf