**Groepsnaam:** Bilthoveniers

**Teamleden:** Malek (Scrum Master & Git Boss), Sebastiaan, Lars, Armin

**Opdrachtgever / Klant:** Annex Bios B.V.

**Project:** Website + reserveringssysteem voor Annex Bios (hoofdkantoor + regiokantoren)

**Wij zijn:** Vestigingslocatie **Bilthoven** (regiokantoor)

# **📋 Plan van aanpak (stappen)**

## **1) Kennismaken & behoefte-inventarisatie**

* Briefing doornemen (Annex Bios: uitbreiding regio + digitaal opschalen).
* Randvoorwaarden vastleggen: responsive, HTML5-validatie, prepared statements, asynchroon laden, Git-flow, online oplevering.
* Aansluiten bij organisatiestructuur:
  + **Utrecht (hoofdkantoor)** beheert TMDB-import + centrale film/tijden-DB + API.
  + **Regiokantoren** (Leerdam, Maarssen, Breukelen, Bilthoven, Montfoort, Woerden, Leidscherijn, Zeist) lezen film/tijden **via HQ API/DB**, slaan **eigen reserveringen** lokaal op.

## **2) Debriefing opstellen en laten accorderen**

* Doel, doelgroep(en), eisen, scope, risico’s en planning samenvatten.
* Specifiek opnemen: datastromen (TMDB → HQ DB → regio’s), beveiliging (inputvalidatie, htmlspecialchars(), **prepared statements**), performance en monitoring.
* Goedkeuring door Annex Bios (product owner/klantcontact).

## **3) Functioneel ontwerp (FO)**

* Pagina’s & user flows: films → tijden → stoelselectie → betaling (mock/placeholder) → bevestiging/mail.
* Rollen: bezoeker (zoeken/boeken), regiobeheer (CRUD locatie + inzage verkoop/reserveringen), HQ-beheer (TMDB-import + tijdschema’s).
* Wireframes desktop & mobiel (zoek, filmlijsten, detail, checkout, beheer).
* Accessibility & UX-richtlijnen (keyboard, contrast, foutmeldingen).

## **4) Technisch ontwerp & ontwikkeling (TO)**

* **Stack & tools:**
  + **PHP** met **Composer**, **Template Engine** (bijv. **Twig** of **Blade**), **MySQL/MariaDB**.
  + **Front-end:** HTML5 (gevalideerd), CSS, JS (fetch/axios) voor **asynchrone** requests.
  + **API’s:**
    - HQ: TMDB-client + normalisatie → **centrale API** voor films/tijden.
    - Regio: **consumeert HQ API**, lokale API/CRUD voor reserveringen + locatiebeheer.
* **Security:** prepared statements, inputvalidatie, htmlspecialchars(), CSRF-tokens voor forms, .env secrets.
* **Git workflow:** feature-branches, pull requests, **Malek** beslist merges naar develop/main.
* **Deployment:** staging + productie (project moet **online**).
* **Logging/Errors:** server logs + nette foutpagina’s.

## **5) Content & koppelingen**

* TMDB-velden mappen naar eigen datamodel (titel, synopsis, poster, runtimes, genres).
* Tijdschema’s per vestiging (HQ beheert; regio leest).
* E-mailbevestiging reservering (transactie-ID, vestiging, zaal, stoel, tijd).

## **6) Testen & revisie**

* Unit/integretietests (API-routes, DB-queries, edge cases stoelreservering).
* Cross-browser & responsive checks.
* Beveiligingstesten (SQLi, XSS, CSRF).
* Gebruikerstesten (happy flow + foutafhandeling).
* Feedbackronde met klant, fixes, her-test.

## **7) Oplevering**

* Final check + technische documentatie + korte beheerhandleiding.
* Deploy productie; monitoring aan.
* Uitleg aan Annex Bios over beheerprocessen (HQ en regio’s).

# **✉️ Debriefing (e-mail aan de klant)**

**Aan:** [contact@annexbios.nl](mailto:contact@annexbios.nl)

**Onderwerp:** Debriefing – Website & Reserveringssysteem Annex Bios

Beste Annex Bios,

Dank voor de briefing. Hieronder bevestigen wij onze interpretatie en aanpak:

**1. Opdrachtgever**

Annex Bios B.V., operator van een bioscoop in Utrecht met lopende overnames in de regio.

**2. Opdracht**

Realiseren van een website met geïntegreerd reserveringssysteem geschikt voor zowel het **hoofdkantoor (Utrecht)** als de **regiokantoren**: Leerdam, Maarssen, Breukelen, Bilthoven, Montfoort, Woerden, Leidscherijn en Zeist.

**3. Doelstellingen**

* Centrale **film- en tijdenbeheer** door HQ met data van **TMDB API**.
* Regiokantoren **consumeren** HQ-data en **slaan eigen reserveringen lokaal op**.
* Eigen **CRUD** per vestiging voor locatiegegevens en verkoop-/reserveringinzicht.
* Responsive, snel en gebruiksvriendelijk platform, veilig en schaalbaar.

**4. Doelgroep**

Bioscoopbezoekers in de regio, HQ- en regiobeheerders (backoffice).

**5. Kwaliteit & Techniek**

* HTML5 (gevalideerd), responsive design, asynchroon dataverkeer.
* Veiligheid: inputvalidatie, htmlspecialchars(), **prepared statements**.
* Git met feature-branches; **Malek** is merge-eindverantwoordelijke.
* Online oplevering (staging + productie).
* Tools: PHP + Composer, Template Engine, MySQL, TMDB API.

**6. Werkwijze**

Scrum (sprints, stand-ups, reviews/retros), versiebeheer via Git, test- & feedbackmomenten per sprint.

Graag ontvangen we akkoord op bovenstaande debriefing, of jullie aanvullingen/wijzigingen.

Met vriendelijke groet,

**Bilthoveniers** (Malek – Scrum Master & Git Boss)

# **🧭 Planning (indicatief, 5 weken)**

## **Week 1 – Kick-off & Debriefing & Functioneel Ontwerp**

**Doelen:** briefing afronden, datastromen vastleggen, debriefing laten accorderen.

**Taken:**

* Malek: debriefing schrijven & afstemming klant.
* Sebastiaan: datamodelschets (HQ films/tijden, regio reserveringen).
* Lars: risico’s & non-functionals (performance, security).
* Armin: toolingvoorstel (Composer, Template Engine, map-structuur).

**Doelen:** user flows, wireframes (desktop/mobiel), API-contract (HQ ↔ regio).

**Taken:**

* Lars: wireframes UX (zoeken, detail, checkout).
* Sebastiaan: API-contracten (endpoints, payloads, validatie).
* Armin: FO documenteren (rollen, rechten, states).
* Malek: review + klantfeedback verwerken.

## **Week 2 – Technisch Ontwerp & Start Build & Features & Integratie**

**Doelen:** project skeleton, database schema’s, TMDB-ingest (HQ-simulatie), regio-consumptie.

**Taken:**

* Armin: PHP project opzetten (Composer, router, controllers).
* Sebastiaan: DB-migraties & **prepared statements** helpers.
* Lars: TMDB-client + normalisatie (mock HQ API).
* Malek: Git-flow bewaken, CI/checks, code reviews.

**Doelen:** reserveringsflow (stoelkeuze), e-mailbevestiging, regiobeheer-CRUD, responsive UI.

**Taken:**

* Sebastiaan: reserverings-API (create/lock seats, transacties).
* Armin: regiobeheer-CRUD (locatieprofiel, rapportage).
* Lars: front-end asynchrone views (lijsten/detail/checkout).
* Malek: security sweep (XSS/CSRF), accessibility, cross-browser tests.

## **Week 3 – Testen, Revisie & Livegang**

**Doelen:** volledige testdekking, performance, bugfixes, documentatie, **online** zetten.

**Taken:**

* Malek: opleverdossier + beheerhandleiding.
* Team: end-to-end tests + klantdemo + productie-deploy.

# **🔧 Technische randvoorwaarden (geconcretiseerd)**

* **HTML5** validatie; **volledig responsive**.
* **Asynchroon** laden (fetch/axios), nette loading- & foutstates.
* **Database/SQL (veilig):** inputvalidatie, htmlspecialchars(), **prepared statements**, beperkte privileges per user, transacties voor seat-lock/booking.
* **API’s:** duidelijke contracten, statuscodes, server-side validation.
* **Template Engine:** Twig/Blade voor views; herbruikbare componenten.
* **Composer:** autoloading, .env configuratie, PSR-4.
* **Git:** feature-branches, PR reviews; **Malek** beheert merges.
* **Hosting:** staging + productie, HTTPS, error logging.

# **🧑‍🤝‍🧑 Rollen & verantwoordelijkheden**

* **Malek** — Scrum Master & Git Boss: planning, ceremonies, code reviews, merges, quality gate.
* **Sebastiaan** — Data & Back-end: schema’s, queries, reserverings-API (prepared statements).
* **Lars** — Front-end & Integratie: UI/UX, async flows, TMDB/HQ-consumptie.
* **Armin** — Platform & Backoffice: Composer-setup, routing, regiobeheer-CRUD.

# **✅ Definition of Done (DoD)**

* Code gereviewd, tests groen, style-checks geslaagd.
* Veiligheid: validatie, htmlspecialchars(), prepared statements aantoonbaar.
* Responsive, toegankelijk, foutafhandeling aanwezig.
* Documentatie (install, .env, deploy, beheer) up-to-date.
* Feature gedemo’d aan klant en geaccepteerd.

# **📈 Risico’s & mitigatie (kort)**

* **Dataconsistentie HQ ↔ Regio:** versietags + healthcheck endpoints.
* **Dubbele boekingen:** seat-lock met TTL + DB-transacties.
* **Rate limits TMDB:** caching & backoff.
* **Security:** pen-test checklist per sprint.

# **🤝 Groepsafspraken**

* **Te laat:** vooraf melden in teamchat + update voortgang.
* **Ziek:** status delen; klein haalbaar taakje oppakken indien mogelijk.
* **Vakantie:** vrij, tenzij achterstand (dan afstemmen).
* **Niets doen:** 3 waarschuwingen → docent informeren.