Endpoint of APIs

user/customer send short code to start

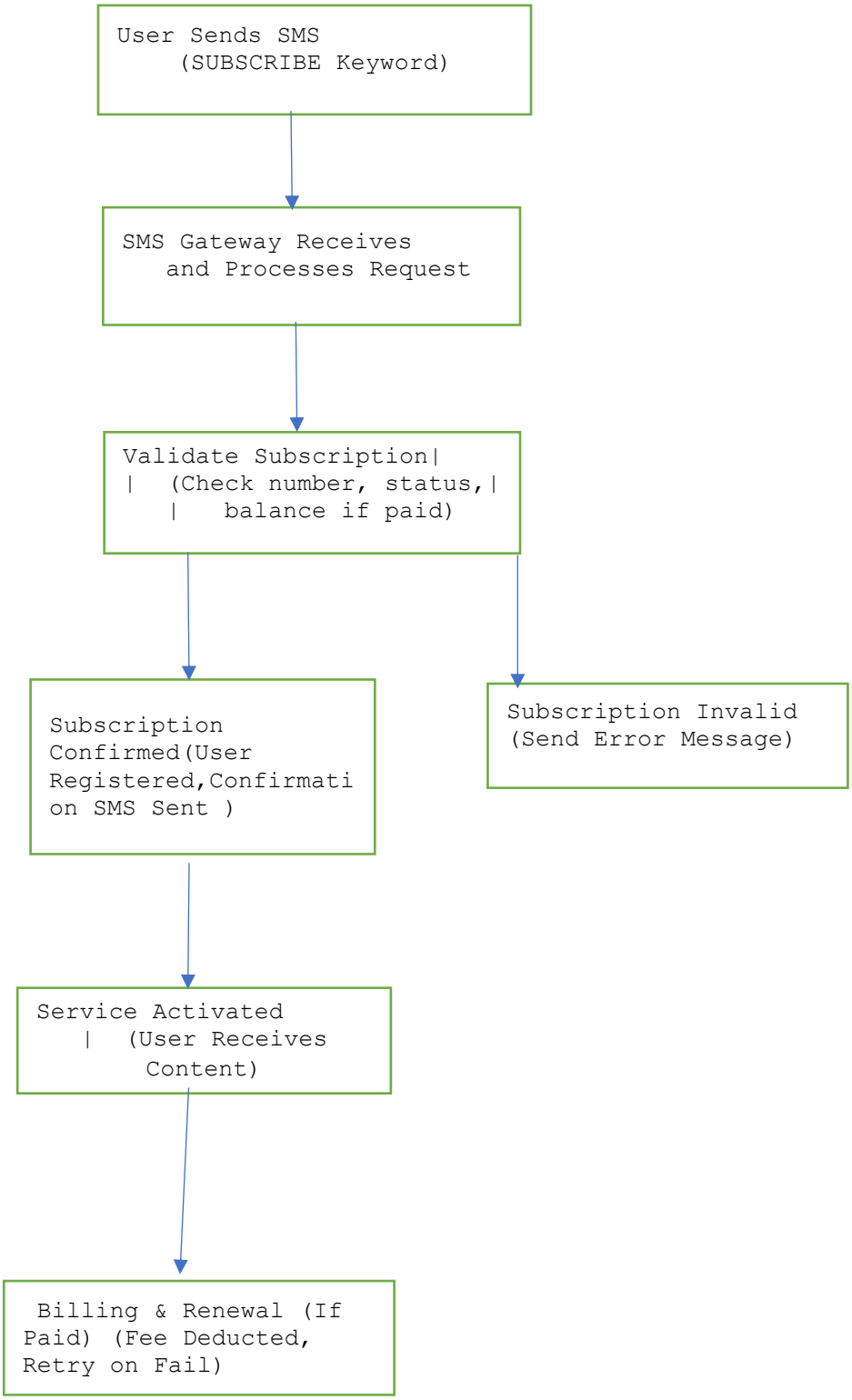
```
  endpoint// Get trending articles
router.get("/trending", getTrendingArticles);
// Create an article
router.post(
  "/",
  authenticate,
  authorize(["admin", "author"]),
  validateSchema(articleSchema),
  createArticle
);

// Update an article
router.put(
  "/:id",
  authenticate,
  authorize(["admin", "author"]),
  validateSchema(articleUpdateSchema),
  updateArticle
);

// Get article by slug
router.get("/slug/:slug", getArticleBySlug);

// Get article by ID
router.get("/:id", getArticleById);

// Delete article
```

```
router.delete(
  "/:id",
  authenticate,
  authorize(["admin", "author"]),
  deleteArticle
);

router.post("/register", validateSchema(registerSchema), register);
router.post("/login", validateSchema(loginSchema), login);
router.post("/logout", logout);
router.post(
  "/",
  authenticate,
  authorize(["admin", "author"]),
  validateSchema(categorySchema),
  createCategory
);

// Get all Categories
router.get("/", getAllCategories);

// Get single Category by ID
router.get("/:id", getCategoryById);

// Get single Category by Slug
router.get("/slug/:slug", getCategoryBySlug);

// Update Category
router.put(
  "/:id",
```

```
  authenticate,
  authorize(["admin", "author"]),
  validateSchema(categoryUpdateSchema),
  updateCategory
);

// Delete Category
router.delete(
  "/:id",
  authenticate,
  authorize(["admin", "author"]),
  deleteCategory
);
// Create Comment
router.post("/", authenticate, validateSchema(commentSchema), createComment);

// Get all comments by article ID
router.get("/article/:article_id", getCommentsByArticleId);

// Get single comment by ID
router.get("/:id", getCommentById);

// Update comment
router.put(
  "/:id",
  authenticate,
  validateSchema(commentUpdateSchema),
  updateComment
);
```

```
// Delete comment
router.delete("/:id", authenticate, deleteComment);

// Create a subscriber
router.post("/", validateSchema(subscriberSchema), createSubscriber);

// Get all subscribers (Admin Only)
router.get("/", authenticate, authorize(["admin"]), getSubscribers);

// Get a single subscriber by ID
router.get("/:id", authenticate, authorize(["admin"]), getSubscriberById);

// Delete a subscriber (Admin Only)
router.delete("/:id", authenticate, authorize(["admin"]), deleteSubscriber);
```

Graphical Representation of Workflow



Graphical Representation of Unsubscription Workflow